

CCAC Capstone Project

May 5, 2009

Client

The client for this project is David Mathews, president and owner of CCAC Air Conditioning and Heating a local small business located at 4502 SPID, Corpus Christi Texas.

Main Users

The main users for this system will be the employees of CCAC, mainly Michelle and Brenda, two of the secretaries. However all employees will have access to the application.

Brief Project Overview

When the project was initially proposed David wanted an application that would help them keep track of maintenance agreement customers. After we accepted the project and met with them to obtain the requirements they also asked for us to keep track of sales leads and filter sales.

By the time we got the requirements all put together the project should handle all of CCAC's customers, maintenance agreements, sales leads, filter sales, financial information, email reminders, as well as generate reports on this information.

Brief Overview of All Major Components of the System

1. Customer

- a. The customer component handles interaction with all data associated to a customer this includes linking customers to other components of the system.

2. Site

- a. The site component handles all information about the location of a maintenance agreement

3. Maintenance Agreement

- a. The Maintenance Agreement component handles agreements that customers have with CCAC for maintenance of equipment.

4. Sales Lead

- a. The sales lead component tracks all possible future sales and what or who generated the future sale.

5. Filter Sales

- a. The filter sales component records all filters sold over the counter or filters used in maintenance agreements.

6. Indoor Air Quality Leads

- a. The indoor air quality leads is a small subset of the sales leads component specifically for leads dealing with indoor air quality (IAQ).

7. Search

- a. The search component handles searching all data in the database so that users can find the data.

8. Reporting

- a. The reporting component generates reports based on data in the database.

9. Email

- a. The email component handles all emails sent by the system.

10. Admin

- a. The admin component handles all administrator functions; these include managing users, managing data in select menus, managing sent emails, email templates, and viewing system logs.

Functional Requirements

The overall functionality of the system is to provide a more efficient method for CCAC to manage their Customer, Maintenance Agreements, Sales Leads, and Filter Sales. Presently these areas are currently being solved through either their Service 1st software, which they are unhappy with, or Microsoft Excel documents. They would like the new system to completely replace the Excel spreadsheets and to compliment the features of their Service 1st software.

The users of our software will be able to add, edit and view details pertaining to customers, customer's sites, maintenance agreements filter sales, and sales leads.

There will be delete functionality of the system but it will not actually delete the information selected. It will just flag the data in the database as deleted and will not come up in the search results. If something was to be deleted then all objects that have a relationship with it would need to be deleted and this is not always desirable.

Non-Functional Requirements

The client side of the system must be developed to work on a pc with at least 256mb ram, 2.08Ghz Processor, running windows XP SP2 and an 800x600 screen resolution. It also needs to be compatible with Windows Vista for future reasons. All client computers must have .Net 3.5 sp1 installed as well as Microsoft Excel 2000+, and MySQL Connection software.

The server side of the system must be developed to work on a Windows Server 2003 running a MySQL database which allows LAN connections.

The new system needs to be efficient with the ability to run all day without abusing resources.

There will be different levels of users in the program. These are broken down into four different user types which are support, admin, clerical and limited user categories. These will be explained further later on.

The system shall include a user authorization procedure where users must identify themselves using a login name and password. Only users who are authorised in this way may access the system data. If a user is to be removed from the system their user information stays in the database. There is then a lock placed on this users log in ability. This lock does not allow them to login. The password of each user is also to be encrypted inside the database for security reasons.

Any user form input errors will be notified and prompted to fix these issues before inserting or updating the database ensuring quality data input.

Any software errors that take place in the system will be logged into the database and will then notify the user that the task has not been completed because of error.

May 5, 2009

Their current software can run queries which are outputted into an excel document which should be parsed and then imported into the new application.

The program needs to be relatively easy to be trained on by having a non complicated graphical user interface to allow employees to achieve independent use of the software quickly. There should also be an instructional video explaining how to accomplish tasks.

The only currency that will be integrated into the system will be American US Dollars.

The system must allow for multiple users to be using the software from different workstations at the same time.

The system must be able to export information and send out emails or text messages with the information.

The system must be able to export reports run on the computer and inputted into a Microsoft Excel file.

Structured Requirements for CCAC Project

Please see Appendix A

Design

Database

Normally with a .Net program one would use MsSQL as the database. However since the license for it is expensive, and there is not any need to use it over the teams preferred choice of MySQL, we went ahead and chose MySQL as the database to run our application.

When designing the database we had many conditions that it had to handle. The biggest was that it had to handle 1:N relationships between many of the tables. When we met to design the database we met in the CCH and had each member bring a design of how they would design the database, and then drew one of the designs up on the window. We then debated for many hours on which way would be best, with input from every member of the team. The initial design of the database had a couple M:N relationships which required those nasty intermediate tables. As we studied the database further as the project went on we slowly realized that we could remove all of these M:N relations and still obtain the desired relationships.

Data redundancy was not an issue in the design. This is because the database has a very large amount of available disk space and the data being stored is pretty small. By having more data redundancy the database would also have a better speed performance by allowing us to avoid the need use as many

May 5, 2009

joins when query for the information were performed. This also enabled the task of writing the queries to be simpler for us, the developers. Since time was not something we had a lot of this helped to speed up development.

Of course with the initial design of anything it does not stay the same way throughout the entire development lifecycle. Every table within the database was changed at some time during the development phase; adding fields or deleting fields was very common once the actual development started taking place. Once developing a section one might see that some functionality was overlooked or elements of something was needed elsewhere for a different form. These were all changes that were transitioned into the database as development continued, but they were never a large problem.

Classes

The classes were designed to make the most out of the Objected Oriented design style. We wanted to be able to reuse as much of the code as possible. For instance we realized that customer, sites, and maintenance agreements all have a Name, Contact, and Address information thus we made each of these a very small class that could be reused in all three instances, saving us from having to write that functionality three different times.

For each tab we had to create a class that was derived from the .net tab class. We had to do this in order to allow multiple instances of the same tabs, and so that they could be created at runtime.

Within these derived tabs we never directly interact with the database. We decided on this to keep a layer of separation to make it so that any changes made to the database did not require changes in multiple files which made the changes easier to make.

The base classes that interacted with the database were set up in such a way that the class would determine if it should update the record or add a new record. This allowed us to have less logic implemented within the classes which interact with the GUI making the slower GUI development slightly easier.

Graphical User Interface

The largest problem faced when designing the GUI was the fact that the software had to run on an 800x600 screen resolution. Many of the designs were about 100-200 pixels to tall and thus we had to make large changes to the overall design many times to get everything to finally fit.

Since each tab class was derived from the .net tab class, we could no longer use the design view within visual studio to make changes to the GUI. This may not sound like a huge problem but there was a time in development when one of the files had over 9000 lines, finding the exact label you are looking for can be quite difficult.

The auto-complete feature for searching for customers or sites was another large challenge to overcome. The main problem with this was that the built in .net auto complete item would only auto-complete if the results matched the first letter of the returned results. Since the client wanted to be

May 5, 2009

able to match any word in the result string this was not an option. Also since our results were in the form of [001] Last, First Middle Street... The user would have to type the bracket and number to get the results they were looking for. Not a very practical solution, however we solved this problem by writing a custom object that was actually a textbox and a list box put together. When the user typed the list box would be placed right below the textbox and show the results in it. Then the arrow keys and selection keys were replaced to give the desired usability.

After much thought about the design of the software we realized that having the auto-complete feature was not good for usability and so we ended up changing the way we handled the relationships between components. As you can see in the prototype GUI on the home page you we had an add button for each section of the software. Each of these tabs had this auto-complete feature that would link the component to a customer. After doing this a question was raised this question was; what happens if there are two John Smiths that live on Alameda? How would the user know which one to pick? The answer is that it would be difficult. Thus we decided to removed all the add buttons, except customer, and made it so that a user could only add these relationships from the customer tab. This would remove any chance of picking the wrong customer, as well as provide a better and easier experience for the user. It may be a few more clicks but the extra quality and reliability should be worth it.

All of the icons used in the project are from the Crystal Project. They are released under the GNU General Public License which gives us pretty much free use for anything. The "Crystal Project exists to create a vivid a consistent experience for the user, through a design delineated by the observation of semiotic, psychological and anthropologic concepts." The icons can be downloaded from their site located at <http://www.everaldo.com/crystal/?action=preview>

Analysis

Use Cases

See Appendix B

Prototype

See Appendix C

Design

Sequence Diagrams

See Appendix D

Database Diagram

See Appendix E

Class Diagrams

See Appendix F

Implementation Details

For this project we needed a language that would work on Windows XP and Windows Vista and be able to connect to, and exchange data with, Microsoft Server 2003 with a MySQL database. The Language must also be friendly when it comes to making GUI's and allow for quick development due to the completion date. These lead us to believe that one of the .Net languages would be the best choice. Since we all had a background in C++ from the university and Cory, Mike and Taylor already had experience with C# we went ahead and chose C# as our main programming language for the project.

Normally with a .Net application one would use SQL Server to run the database because of the way the .Net languages are built to interact with it. However this was not a choice for us since the client did not wish to spend the money on a license for it. This led us to use MySQL as it is the leading open source database and the team already had experience some with it.

Testing Activities

To test the program we would have members of the team be the testers and make sure it worked as it should. The best way we found to do this was to have each member test a section they did not write. No one wants to give them self more work, so if we were to test our own section it will almost always work correctly. However if we test someone else's section then they have to fix it and it will be tested better more thoroughly. We did this repeatedly as we went through development of the software and took note of any bugs that came up and added them to a "todo" document.

Meeting Minutes

See Appendix G

Major Code Components

See Appendix L

User Manual

See Appendix M

May 5, 2009

Known Issues

See Appendix I

Recommended Future Work

See Appendix J

Time Sheet

See Appendix K

Appendices

Appendix A - Structured Requirements
Appendix B - Use Cases
Appendix C - Prototype
Appendix D - Sequence Diagrams
Appendix E - Database Diagrams
Appendix F - Class Diagrams
Appendix G – Meeting Minutes
Appendix H – Status Reviews
Appendix I – Known Issues
Appendix J – Recommended Future Work
Appendix K – Time Sheet
Appendix L – Major Code Components
Appendix M – User Manual

Structured Requirements

Appendix A

May 5, 2009

The following is a detailed listing of all requirements for the CCAC project. All listed features will be developed into the final software package. Any changes to the project should be made here as well as any other needed places.

Contents

Structured Requirements for CCAC Project	1
System Requirements	12
User Login Credentials	13
Customer Information	13
Technician Information	14
Site Information	14
Maintenance Agreements	14
Sales Leads	16
Filter Sales	18
Indoor Air Quality (IAQ) Leads	18
Search	18
Reports Module	19
Emails	19
Admin	19

System Requirements

May 5, 2009

- a. Windows XP SP2 and SP3, Windows Vista
- b. Windows Server 2003 Small Business
- c. MySQL 5.x
- d. .NET 3.5 SP1
- e. C#.NET
- f. LAN Connections to server

User Login Credentials

- a. Login is needed to access software
- b. Users will be one of four levels.
 - i. Support
 - ii. Admin
 - iii. Clerical
 - iv. Limited
- c. Each level has different levels of access which will be stated throughout the document
- d. Creating and editing users will be defined in the admin section of this document

Customer Information

- a. All customers are input into the database using the customer input form. These customers will then be connected with other sections of the system.
- b. Data Collected on Each Customer
 - 1. Customer ID Number, Unique User input number
 - 2. Old Customer ID Number, from previous software (User input value)
 - 3. Associated Site Number (If Available) (Multiple Entries Available)
 - 4. Primary Customer Information
 - i. Name Prefix (Dropdown List: Mr., Ms., etc.)
 - ii. First Name
 - iii. Middle Initial
 - iv. Last Name
 - v. Name Suffix (Dropdown List: Jr., Sr., etc.)
 - vi. Address Street Number
 - vii. Address Street Name
 - viii. Address Line 2
 - ix. City
 - x. State
 - xi. Extended Zip 00000-0000
 - xii. Home Phone
 - xiii. Cell Phone
 - xiv. Work Phone
 - xv. Fax
 - xvi. Email
- b. Alternate Customer Information (Same as Above--if Available)
- c. Comfort Advisor (Dropdown List)
- d. Preferred Technician (Dropdown List: All Technicians)

May 5, 2009

- e. Special Notes Field (Large Multi-line Text Box)

Technician Information

- a. All information about Technicians will be input and modified using the technician form.
- b. Input and Update is limited to Support, Admin, Clerical user levels
- c. Technician Information Collected:
 - 1. First Name
 - 2. Last Name
 - 3. SAME NAME INFO FOR COMFORT ADVISORS(CA)

Site Information

- a. Each job site will be input and updated using the job site form.
- b. Information Collected:
 - 1. Unique Site Number
 - 2. Contact First Name
 - 3. Contact Last Name
 - 4. Contact Phone
 - 5. Alternate Contact Phone
 - 6. Contact Email
 - 7. Address Street Number
 - 8. Address Street Name
 - 9. Address Line 2
 - 10. City
 - 11. State
 - 12. Extended Zip 00000-0000
 - 13. Site Notes

Maintenance Agreements

- a. This section will manage maintenance agreements. These maintenance agreements should be associated with a customer ID number, and a site number if available.
- b. History of maintenance performed should be kept in log.
- c. If a sales lead is generated from a maintenance agreement then that sale should be linked. And track \$ amount of Sale
- d. Data input to be stored for each agreement
 - 1. Customer ID Number
 - 2. Associated Site Number (if available)
 - 3. Last Time Contacted (Store Entire History of Contacts connected to each Inspection Notation (Spring 09, etc.)
 - ii. Input data field for Type of Contact.
 - 1. Email (Info with date sent)
 - 2. Post Card (Info with date sent)
 - 3. Phone (Include Text box for notes)
 - 4. Contract Information

May 5, 2009

- iii. Status (Dropdown List: Active, Expired, Potential)
- iv. Start Date
- v. Expiration Date
- vi. Price of Agreement
- vii. Payment Date
- viii. Number of Inspections
- ix. Starting Inspection Notation: (Such as Spring 09 or Fall 09, etc.) System will automatically figure all inspections due based on Number of Inspections.
- x. Number of Systems
- xi. Number of Dehumidifiers
- xii. Filter Type
- xiii. Travel Amount \$
- xiv. Extra for Filters \$
- xv. Misc. Extra Description
- xvi. Misc. Extra \$
- xvii. Salesperson (Dropdown List: All Personnel)
- xviii. Spiff Amount
- xix. Type of agreement (Dropdown List of the Following)
 - 1. Prepaid
 - 2. Monthly Billing
 - 3. Billed Time of Service
- xx. Equipment Information – Allow for Multiple
 - 1. Area Equipment Handles
 - a. T-Stat type and Model:
 - b. AC-Enhancer Model:
 - c. Dehumidifier Model: (Multiple Listings)
 - d. Inside Equipment
 - i. Brand
 - ii. Model Number
 - iii. Serial Number
 - iv. Size
 - v. Year Installed
 - vi. Type (Dropdown list: Gas, Electric, Heat Pump)
 - vii. Orientation (Dropdown list: Upflow, Horizontal)
 - viii. Extended Warranty Type
 - ix. Filter Quantity and Size:
 - x. Extended Warranty Number
 - xi. Extended Warranty Expiration Date
 - xii. Notes (Multi-line Text Data Field)
 - e. Outside Equipment
 - i. Brand
 - ii. Model Number
 - iii. Serial Number
 - iv. Size
 - v. Voltage and Phase:
 - vi. Year Installed
 - vii. Type (Dropdown list: Gas, Electric, Heat Pump)

May 5, 2009

- viii. Location (Dropdown list: Ground, Roof, Platform, Balcony)
- ix. Extended Warranty Type
- x. Extended Warranty Number
- xi. Extended Warranty Expiration Date
- xii. Notes (Multi-line Text Data Field)
- e. Data Entry for Calls that have been run.
 - i. Service Invoice Number
 - ii. Tech Performing Call
 - iii. Inspection Notation (spring 09, etc.)
 - iv. System to Prompt Amount of Revenue to be recognized.
 - v. Any additional repair revenue.
 - vi. Was a Sales Lead Generated Y/N
 - 1. If Y then Prompt for Comfort Advisor Dropdown
 - vii. Equipment Sales Amount. (Tied to Inspection Notation)

Sales Leads

- a. Section for leads on possible service opportunities, as well for tracking the salesmen and how successful they are.
- b. Data Collected will be:
 - i. Lead Type (Dropdown List of the Following)
 - 1. Equipment Replacement
 - a. Complete
 - b. Partial
 - 2. New Construction
 - 3. Commercial
 - 4. Home Upgrade
 - 5. Insulation
 - 6. Dehumidifier
 - 7. IAQ
 - 8. Off Season Work
 - ii. Lead Source (Dropdown List of the Following)
 - 1. Self-Generated
 - 2. Service Lead (Dropdown List of Technicians)
 - 3. Maintenance Agreement Lead (Automatically Prompt for Customer Name to automatically propagate the customer info.
 - 4. Phone (Dropdown List of the Following)
 - a. Yellow Pages
 - b. Television
 - c. Radio
 - d. Newspaper
 - e. Referral
 - f. Other
 - iii. Salesman (Dropdown List CA)
 - iv. Customer (With all info)

May 5, 2009

- v. Date Taken
- vi. Time Taken
- vii. Call Receiver
- viii. Existing Customer (Yes/No)
- ix. Are They Down (Yes/No)
- x. Have PESA (Yes/No)
- xi. System Needs
 - 1. Complete System
 - a. Size
 - b. Horizontal or Vertical
 - 2. Condenser
 - a. Size
 - b. Location (Ground/Roof/Other)
 - 3. Air Handler/Furnace
 - a. Size
 - b. H or V
 - 4. Indoor Air Quality Issues: (Dropdown List able to select multiples)
 - a. Dust
 - b. Humidity
 - 5. New Install Equipment and Ductwork
 - a. Square Foot
 - b. Attic (Y/N)
 - c. Number of Rooms
 - 6. Dehumidifier
 - 7. Appointment
 - a. Date
 - b. Time
 - c. Proposal Signed Date
 - 8. Transmit Options: (Log Communication Dates)
 - a. Text Page Basic Info to CA Phone
 - b. Email Info to CA.
 - c. Print Info
 - 9. Lead Status
 - a. Sold
 - i. Date Sold
 - ii. Job Amount
 - b. Pending
 - i. Proposal Date
 - ii. Follow Up Date
 - iii. Follow Up Contact Records with Notes
 - c. Lost
 - i. Reason
 - 1. Price
 - 2. Schedule
 - 3. Salesman
 - 4. Unknown

May 5, 2009

Filter Sales

- a. Associate Customer with Filter (Size and Quantity Multiple Listings)
- b. Set Filter Reminder Time
- c. Track Filter Purchases (Keep Complete History)
 - i. Date
 - ii. Quantity and Size
 - iii. \$ amount
- d. Date Last Purchased plus Reminder Time triggers Reminder List
- e. Email 'replace filters' reminder to customers.
 - i. Explained in Email Section of this Document

Indoor Air Quality (IAQ) Leads

- a. Type of Lead:
 - i. Current
 - ii. Future
- b. Dust
- c. Mold
- d. Comfort
- e. Duct Repair
- f. Insulation
- g. AEP

Search

- a. Include Option to Search by the Following Terms (Rough List: Some options may need to be added or removed)
 - i. First Name and/or Last Name
 - ii. Customer ID Number
 - iii. Customer Address (also searches Site Addresses)
 - iv. Customer Street Name
 - v. Customer Zip Code
 - vi. Customer Phone Number
 - vii. Customer Email Address
 - viii. Site ID Number
 - ix. Site Contact First and/or Last Name
 - x. Site Contact Phone Number
 - xi. Site Contact Email Address
 - xii. Warranty or Maintenance Agreement Number
 - xiii. Maintenance Agreement Expiration by Date Range
 - xiv. Maintenance Agreement Potential by Date Range
 - xv. Filter Sales Reminder Time by Date Range
 - xvi. Equipment Serial Number

May 5, 2009

Reports Module

- b. Reports should be kept for (Reports to be run, not saved)
 - i. Maintenance Agreements Search Requirements
 - 1. Income Reportable (By Date Range)
 - 2. Renewal Customers (By Date Range)
 - 3. Zip Code
 - 4. Income not yet Reportable
 - 5. Total Income (By Date Range)
 - ii. Filter Sales
 - 1. Filters sold list by date range
 - 2. \$ Sold by date range
 - 3. Customer Reminders due by date range
 - 4. Customer response to reminders.
 - iii. Sales Leads (All Reports by Date Range)
 - 1. Total Leads
 - 2. Total Leads from each source
 - 3. Total Sales from Leads also broken down by source.
 - 4. Total Sales vs. Leads by Salesman (CA)
 - 5. \$ Sold by CA
 - 6. \$ Sold by Lead Type
 - iv. Indoor Air Quality (IAQ) Leads
 - 1. IAQ Leads to be scheduled (Waiting List)
 - 2. IAQ Sales by date range

Emails

- c. Email notices should be queued to be sent to all customers that have an upcoming maintenance to schedules, or have IAQ testing coming. (By Date Range)
- d. Emails should be short and customizable.
- e. If customer does not have email address on file, add notification to list to be printed.
- f. Sent emails should be logged for record keeping.

Admin

- g. Users
 - i. Add Users
 - ii. Edit Users
 - iii. Reset Passwords
 - iv. Lock user accounts
- h. Edit Dropdown Fields
 - i. Name Prefix
 - ii. Name Postfix
 - iii. Maintenance Agreement Status
 - iv. Types of Maintenance Agreement Status
 - v. Comfort Advisors
 - vi. Service Techs

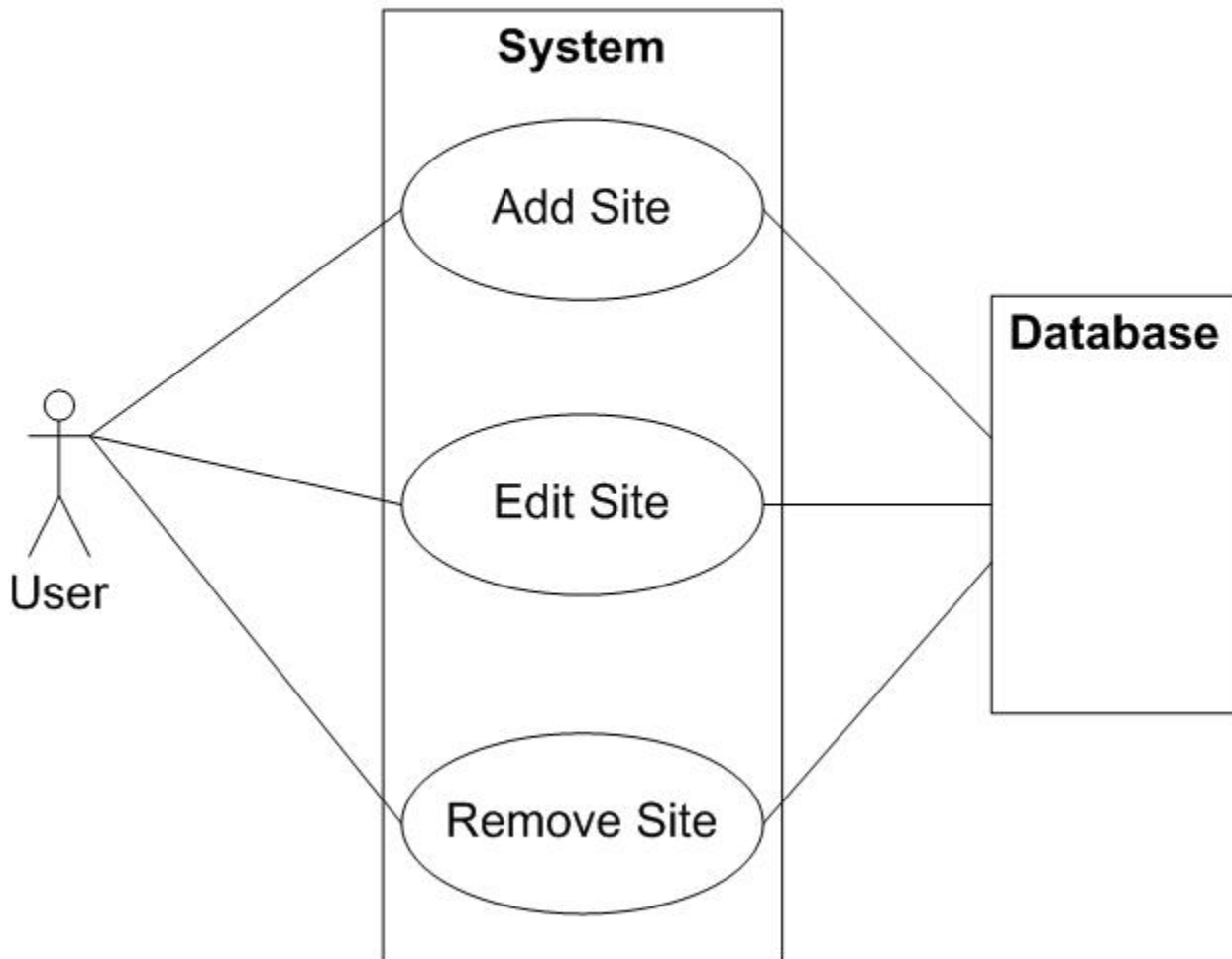
May 5, 2009

- vii. Employee List
- viii. Inside Equipment Type
 - ix. Inside Equipment Orientation
 - x. Outside Equipment Type
 - xi. Outside Equipment Location
- xii. Sales Lead Type
- xiii. Sales Lead Type – Equipment Replacement Sub Menu
- xiv. Lead Sources
 - xv. Sales Lead Phone Sub Menu
- xvi. Filter Types
- i. Manage who gets which emails
- j. Manage Email Message Templates
- k. View Logs

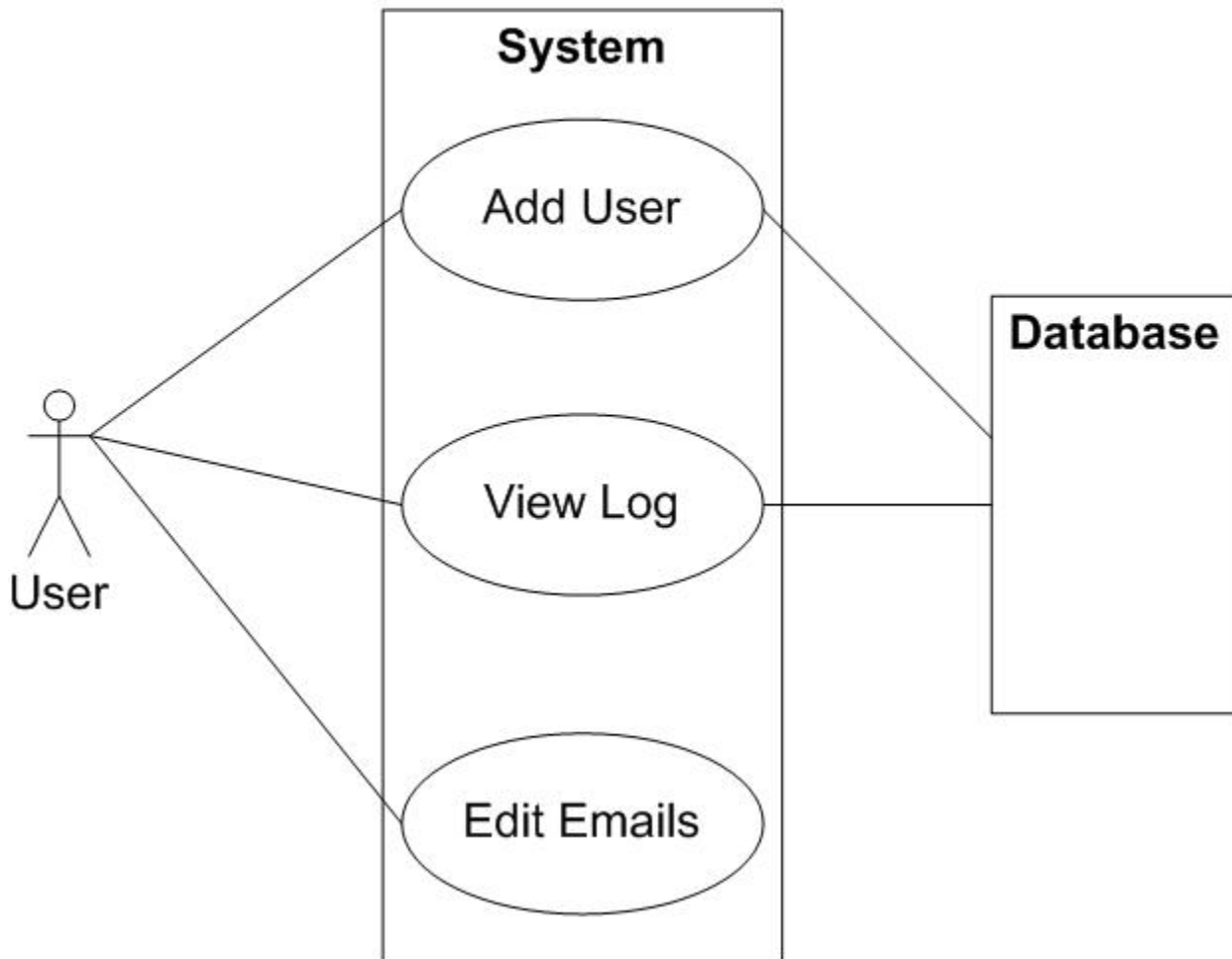
Use Cases

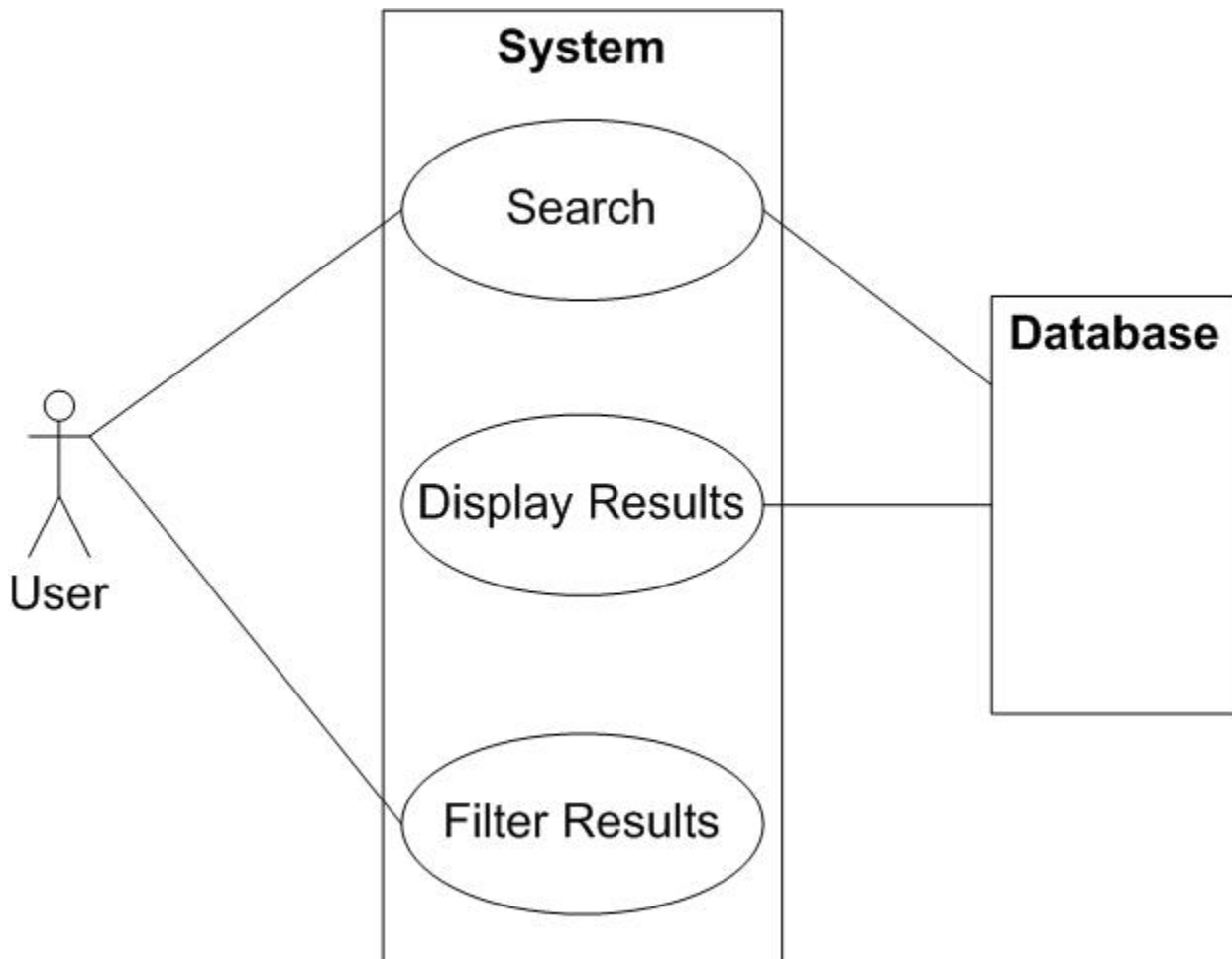
Appendix B

Add/Edit/Remove Site

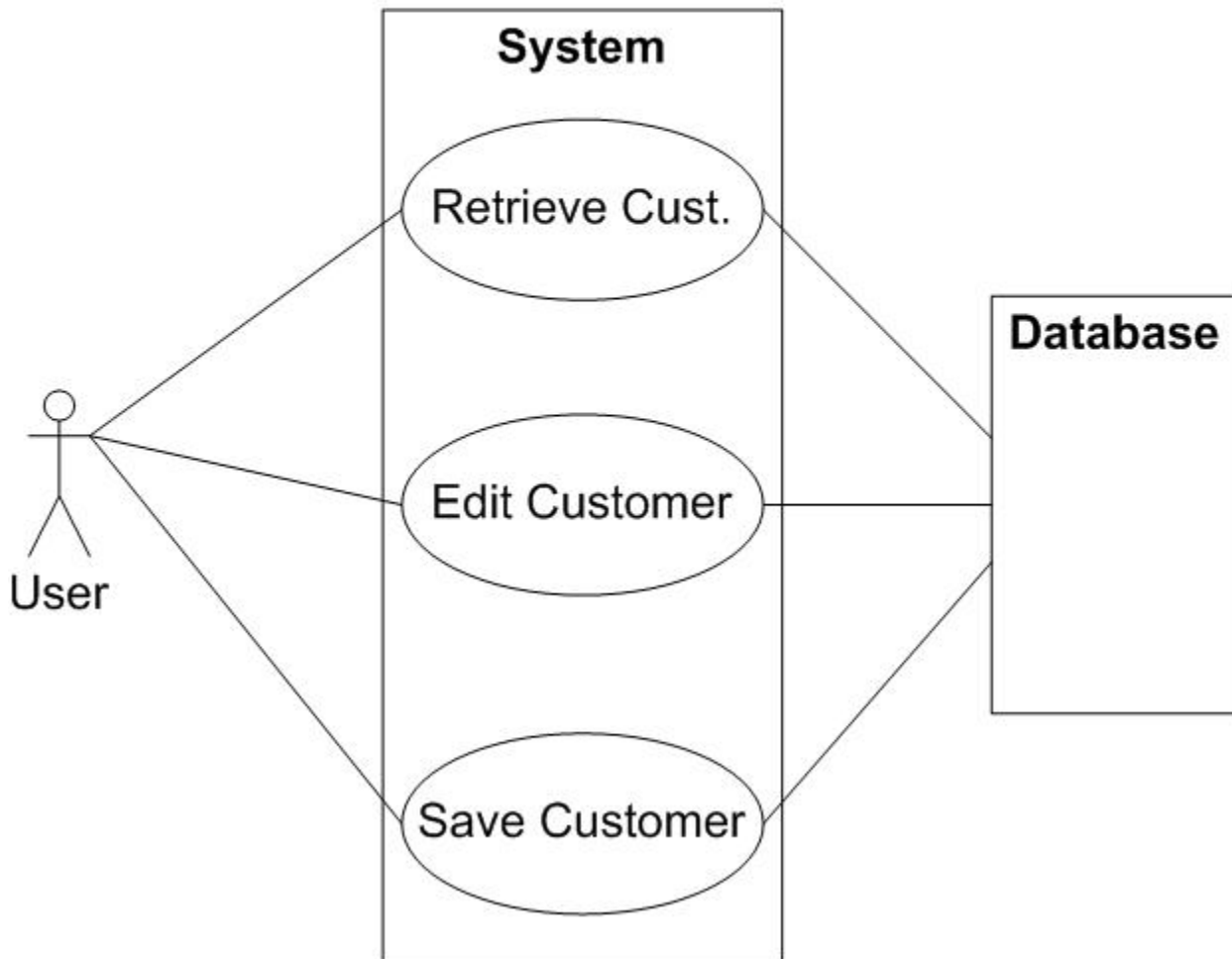


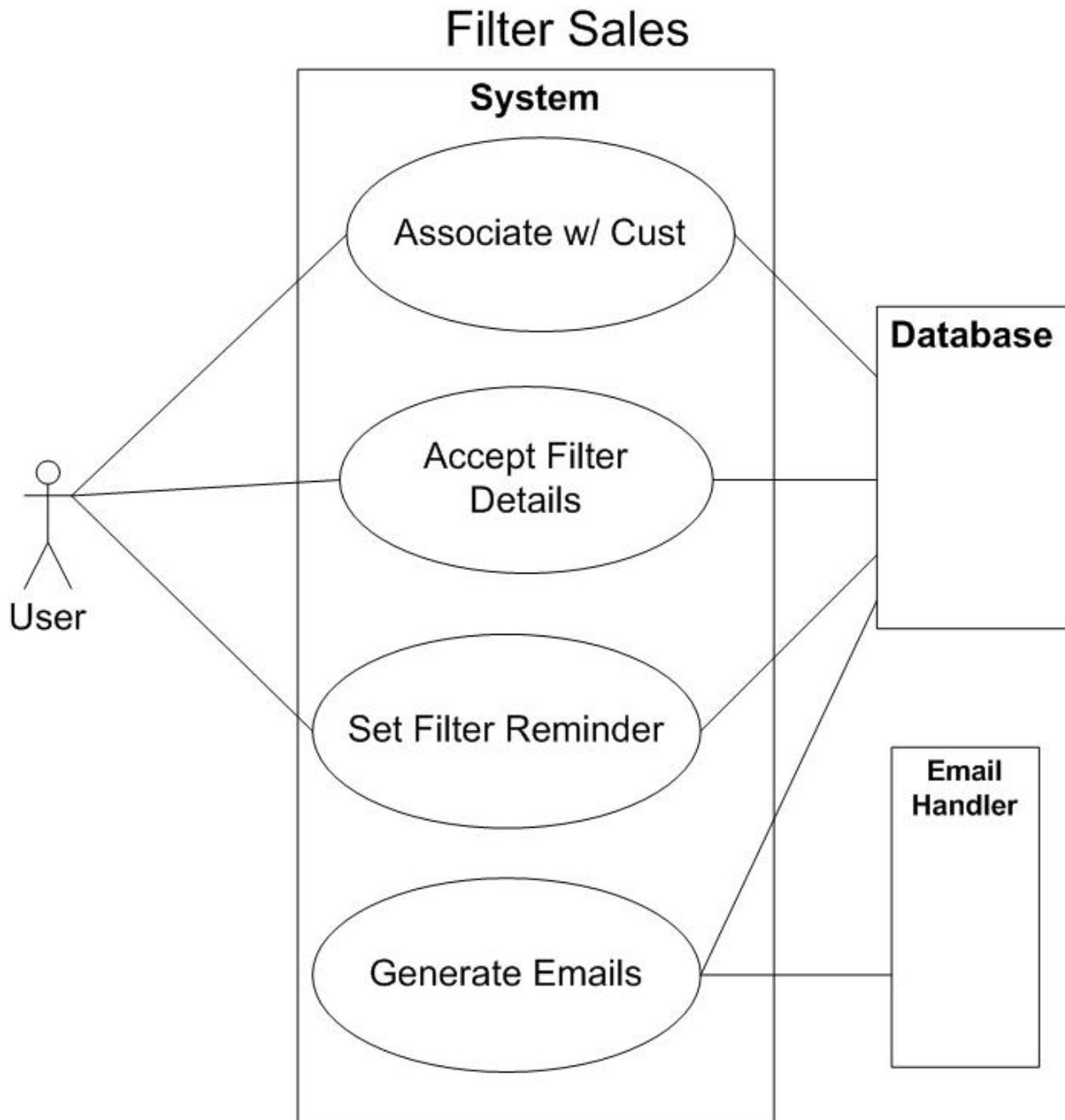
Administrative Duties



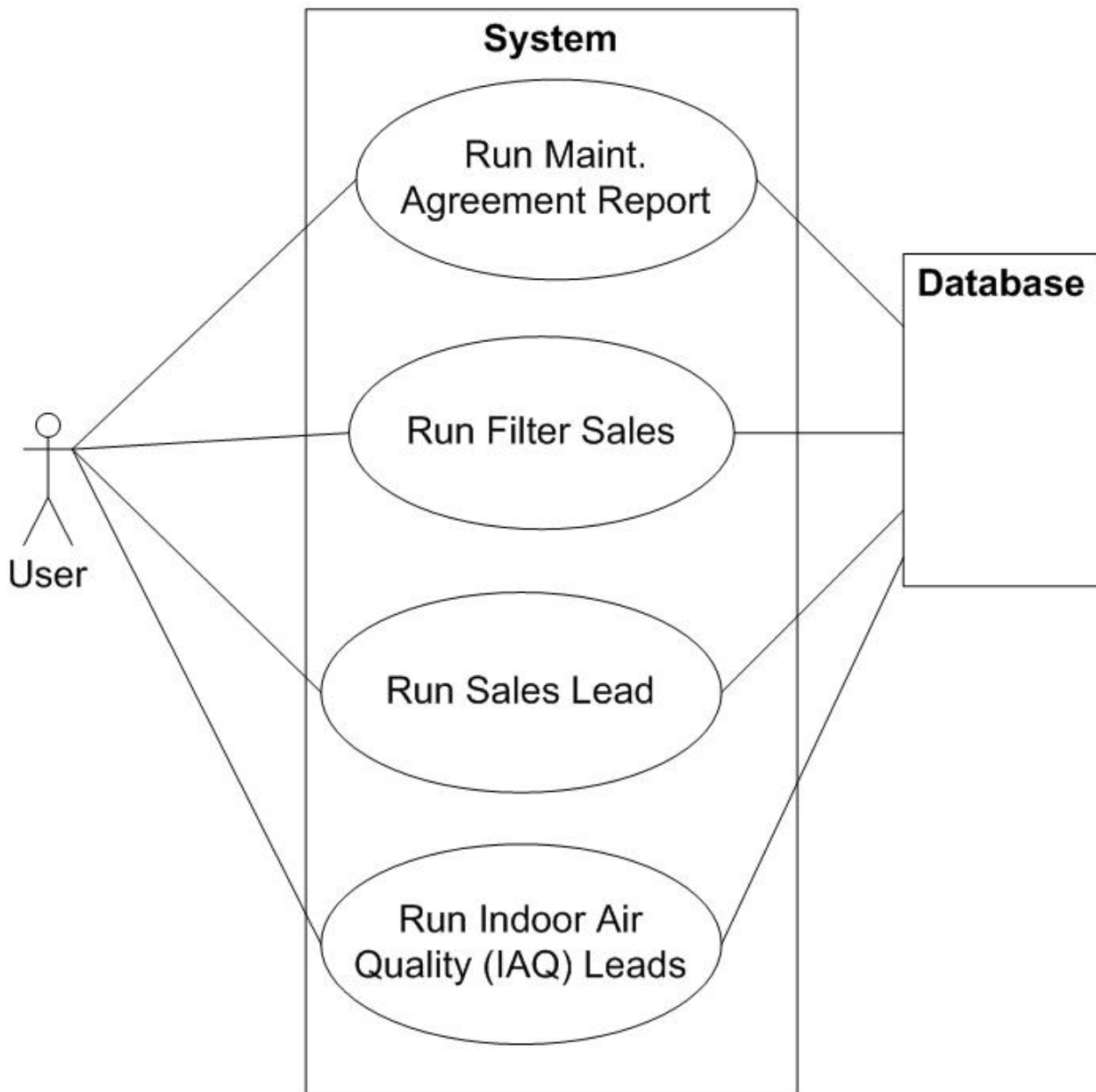


Add/Edit Customer

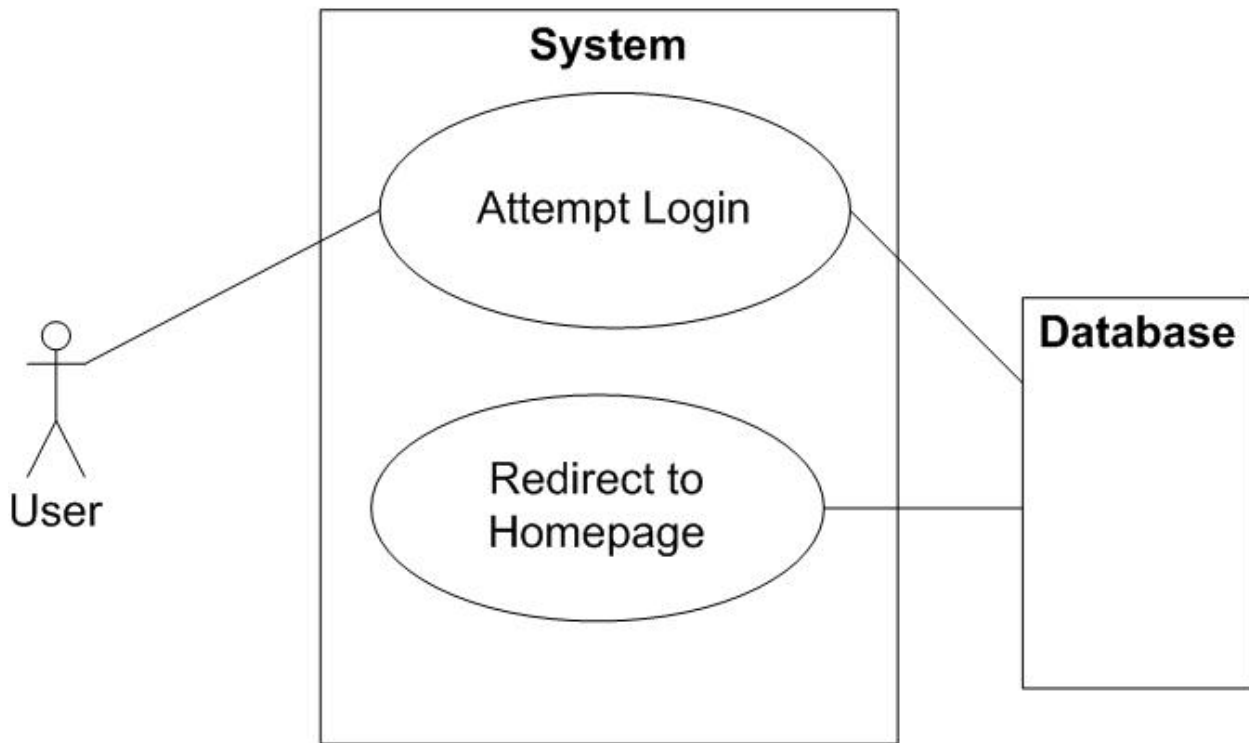




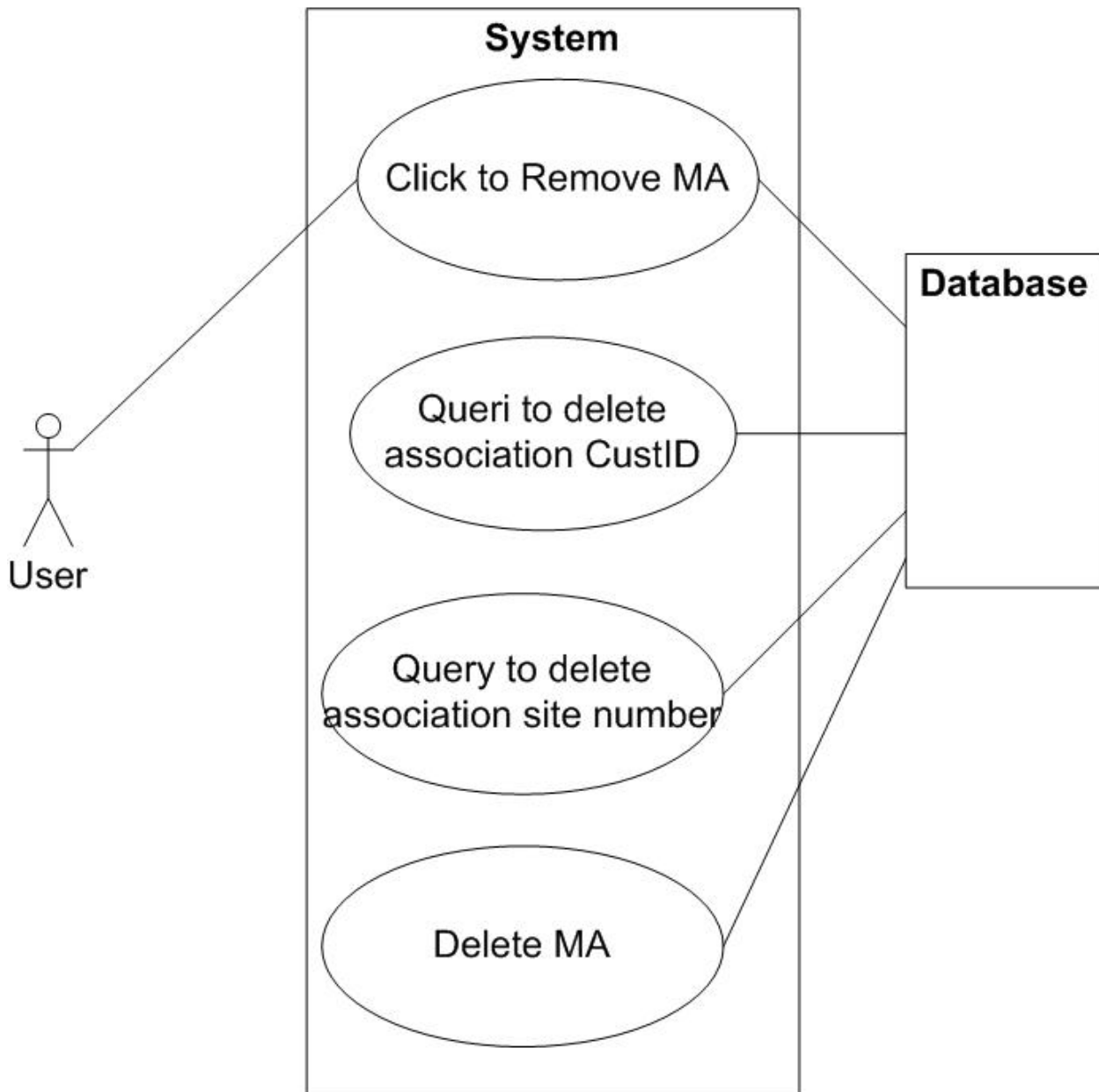
Reports



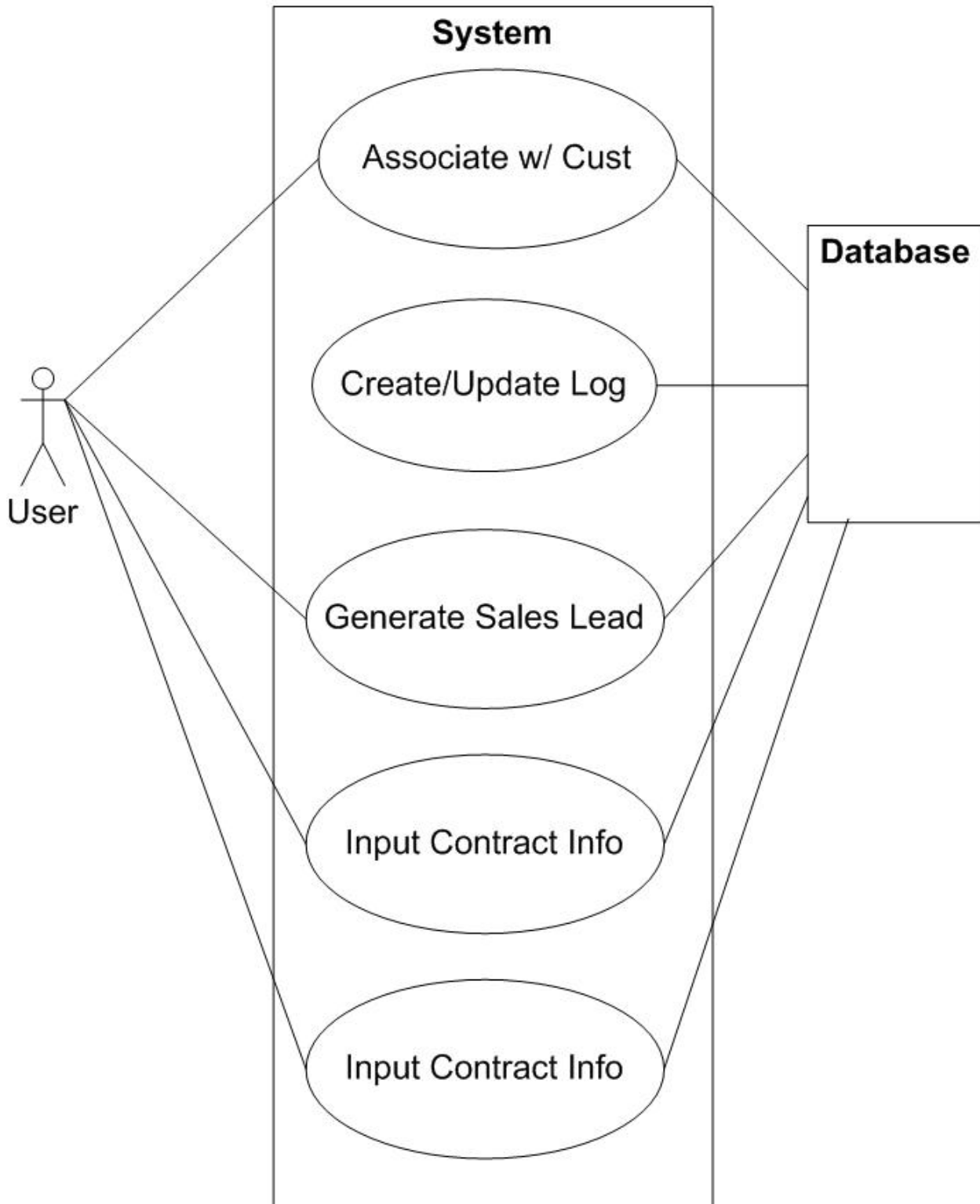
User Login



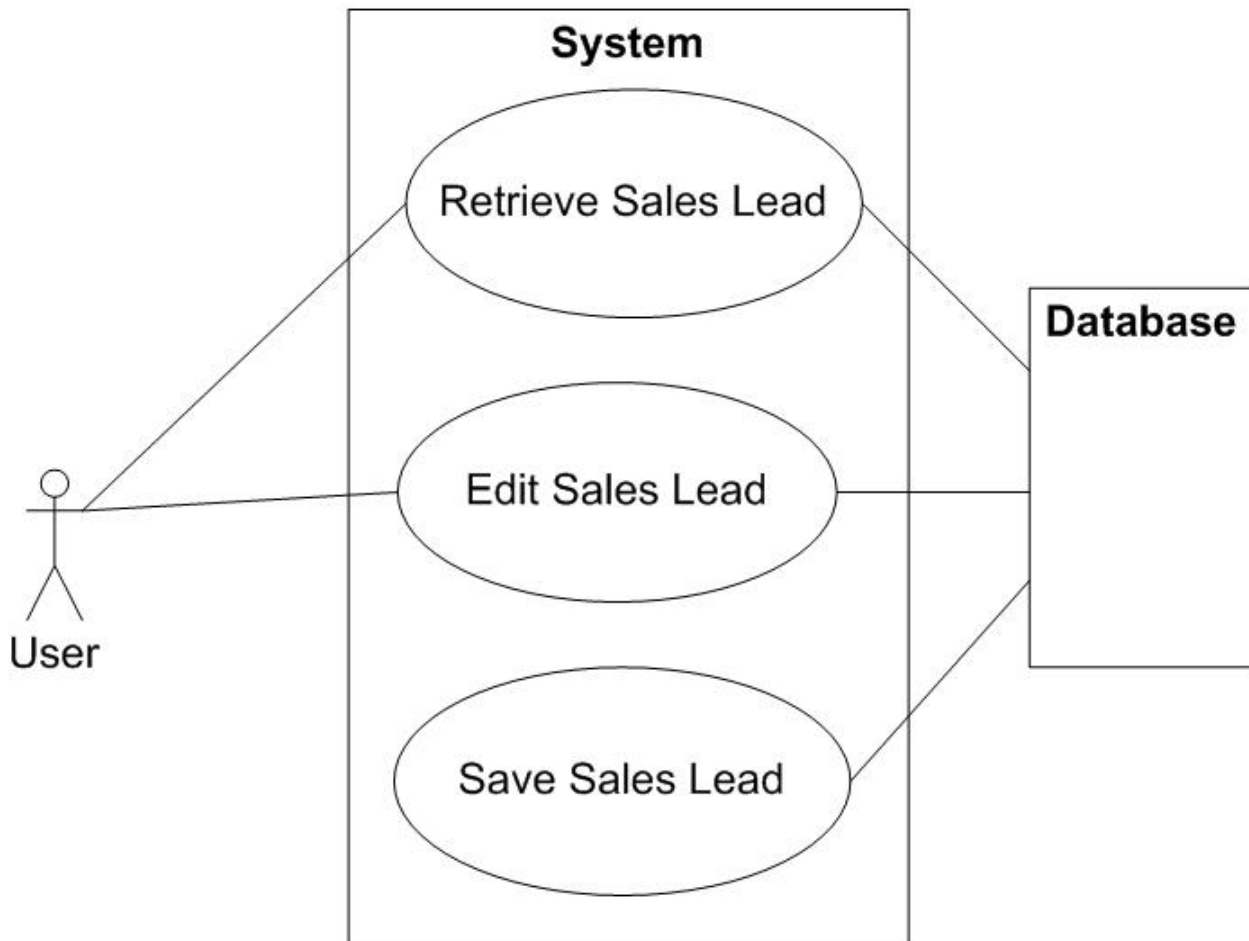
Remove Maintenance Agreement



Maintenance Agreement



Add/Edit Sales Lead



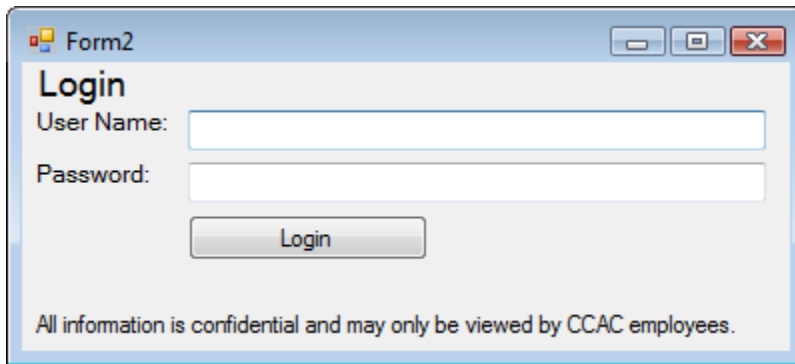
May 5, 2009

Prototype Screenshots

Appendix C

Log in screen

May 5, 2009



Form2

Login

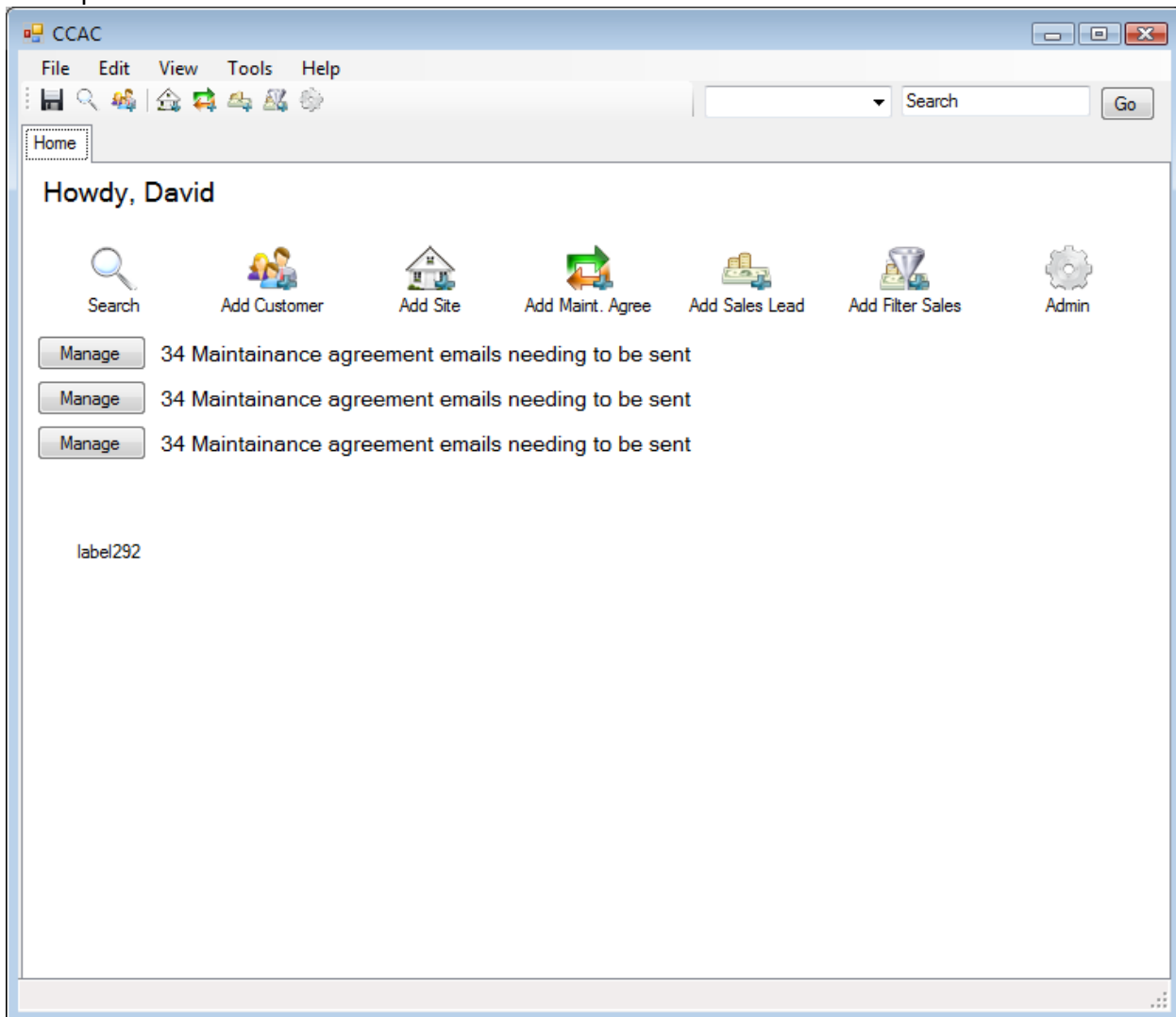
User Name:

Password:

Login

All information is confidential and may only be viewed by CCAC employees.

Startup Screen



CCAC

File Edit View Tools Help

Search Go

Home

Howdy, David

Search Add Customer Add Site Add Maint. Agree Add Sales Lead Add Filter Sales Admin

Manage 34 Maintenance agreement emails needing to be sent

Manage 34 Maintenance agreement emails needing to be sent

Manage 34 Maintenance agreement emails needing to be sent

label292

May 5, 2009

Customer Tab

The screenshot shows a web application window titled "CCAC" with a menu bar (File, Edit, View, Tools, Help) and a toolbar. Below the toolbar is a navigation bar with tabs: Home, Search Results, Customer Overview (selected), Filter Sales, Maintenance Agreements, Sales Leads, Reports, and Admin. A search bar with a "Go" button is also present.

The main content area displays the customer profile for "[001] Smith, John". A button labeled "View Alternate Contacts" is located below the name. The profile is divided into several sections:

- Customer Info:** A form with fields for Name Prefix (Mr.), First Name (John), Middle Initial (J.), Last Name (Smith), Suffix, Address Street (123 Street), Address Line 2 (#4), City (Corpus Christi), State (Texas), Zip (78415 - 1111), Home Phone (361-222-2222), Cell Phone (361-111-1111), Work Phone (361-333-3333), Fax (361-444-4444), and Email (John@Smith.com).
- Comfort Advisor:** David
- Technician:** David
- Recent Filter Sales:** A table with columns "Date" and "Number", listing sales from 2009-02-10 to 2008-12-15. Each entry has a "View" button.
- Current Maintenance Agreements:** An empty box.
- Notes:** A large empty text area.

Date	Number	Action
2009-02-10	2	View
2008-12-15	3	View
2008-12-15	3	View
2008-12-15	4	View
2008-12-15	5	View

May 5, 2009

Search Screen

The screenshot shows a web application window titled "CCAC". The menu bar includes "File", "Edit", "View", "Tools", and "Help". Below the menu is a toolbar with icons for file operations and a search bar with a "Go" button. The main content area has tabs for "Home" and "Search Results", with "Search Results" being the active tab. Under the "Search" heading, there is a dropdown menu set to "Customer" and a text input field containing "Search". A "Search" button is to the right of the input field. Below this, the text "Customer Search Results for : SearchTerm" and "Results 1-3 of 3" is displayed. Three search results are listed, each with a "Select" button to its left. The results are identical: "[001] Mr. John J. Smith" followed by "123 Street #4 Corpus Christi Texas 78415-1111". The second result is highlighted with a light gray background.

Search Results
[001] Mr. John J. Smith 123 Street #4 Corpus Christi Texas 78415-1111
[001] Mr. John J. Smith 123 Street #4 Corpus Christi Texas 78415-1111
[001] Mr. John J. Smith 123 Street #4 Corpus Christi Texas 78415-1111

May 5, 2009

Maintenance Agreement

The screenshot shows a web application window titled "CCAC". The menu bar includes "File", "Edit", "View", "Tools", and "Help". Below the menu is a toolbar with icons for file operations and a search bar with a "Go" button. The breadcrumb trail shows: Home > Search Results > Customer Overview > View Filter Sales > Maintenance Agreements > Sales Leads > Reports > Admin > Add Customer > Ac > >. The main content area is titled "Maintenance Agreement for : Smith, John". It contains two tabs: "Information" (selected) and "Systems". The "Information" tab displays the following fields:

Customer	John Smith	Site	Site A
1435 Evergreen Terrace Corpus Christi, Tx 78412 2326 S Alameda Corpus Christi, Tx 78412		Status	<input type="text"/>
		Start Date	<input type="text"/>
		Expiration Date	<input type="text"/>
		Price	<input type="text"/>
		Payment Date	<input type="text"/>
		Num Inspections	<input type="text"/>
		Starting Inspection	<input type="text"/>
		Systems	<input type="text"/>
		Filter Type	<input type="text"/>
		Salesperson	<input type="text"/>
		Spiff Amount	<input type="text"/>
		Agreement Type	<input type="text"/>

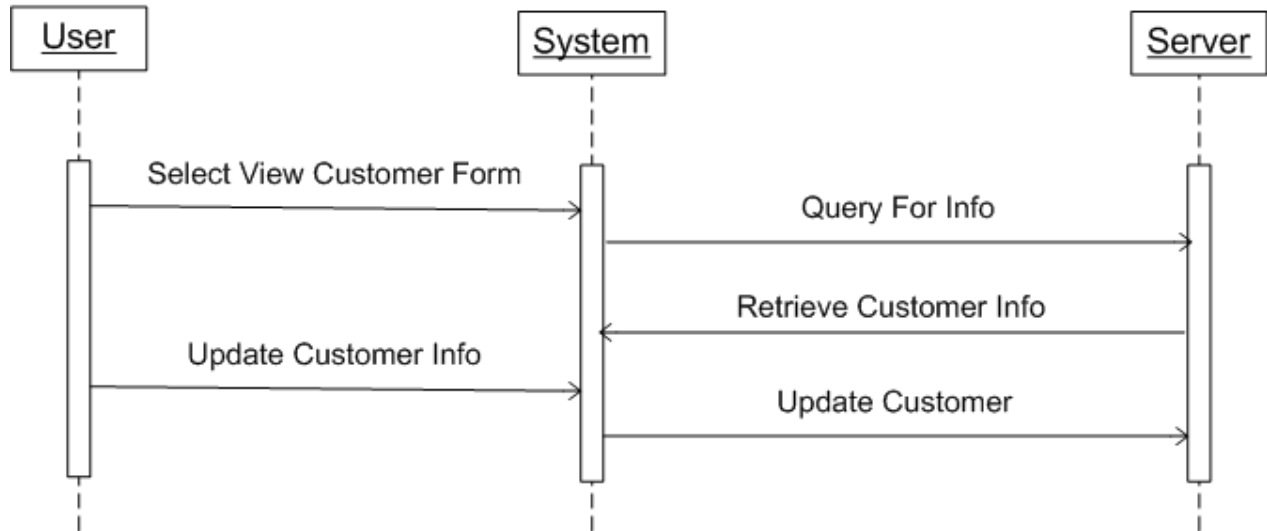
Below the "Information" tab is a "Misc Info" section with sub-tabs: "Misc Info" (selected), "Sales Leads", "Contact History", and "Costs". The "Misc Info" sub-tab contains a "Misc Information" field with a large text area.

Sequence Diagrams

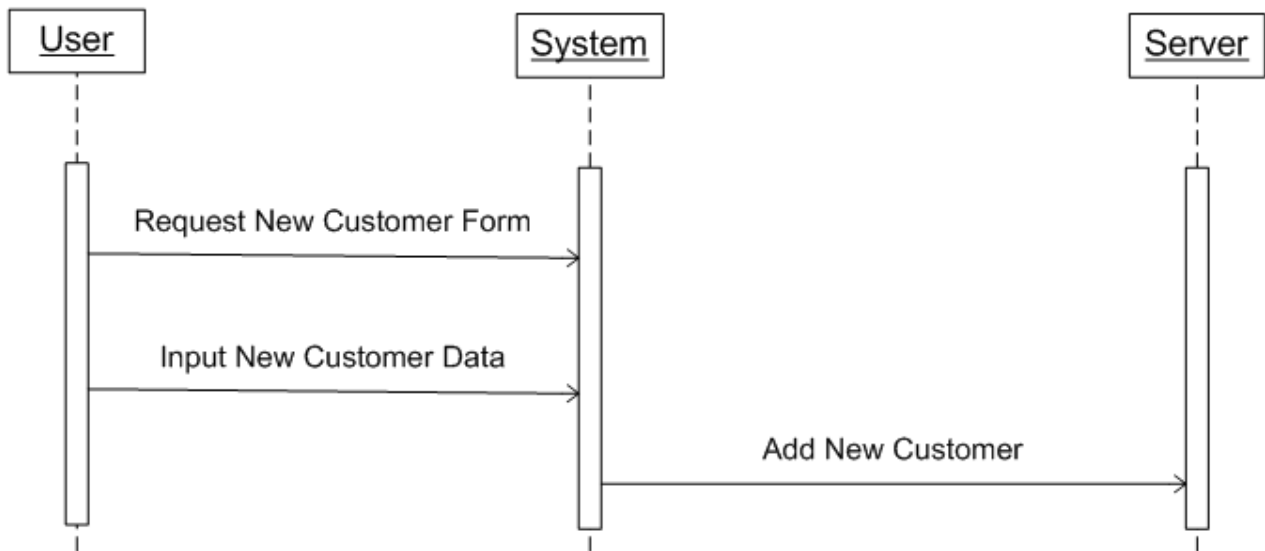
Appendix D

May 5, 2009

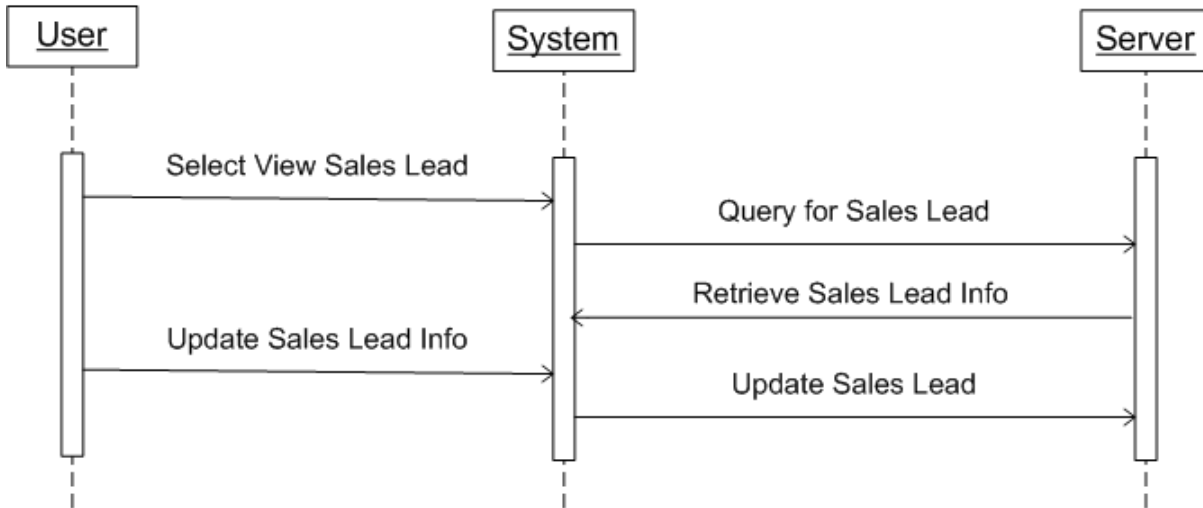
Edit Customer



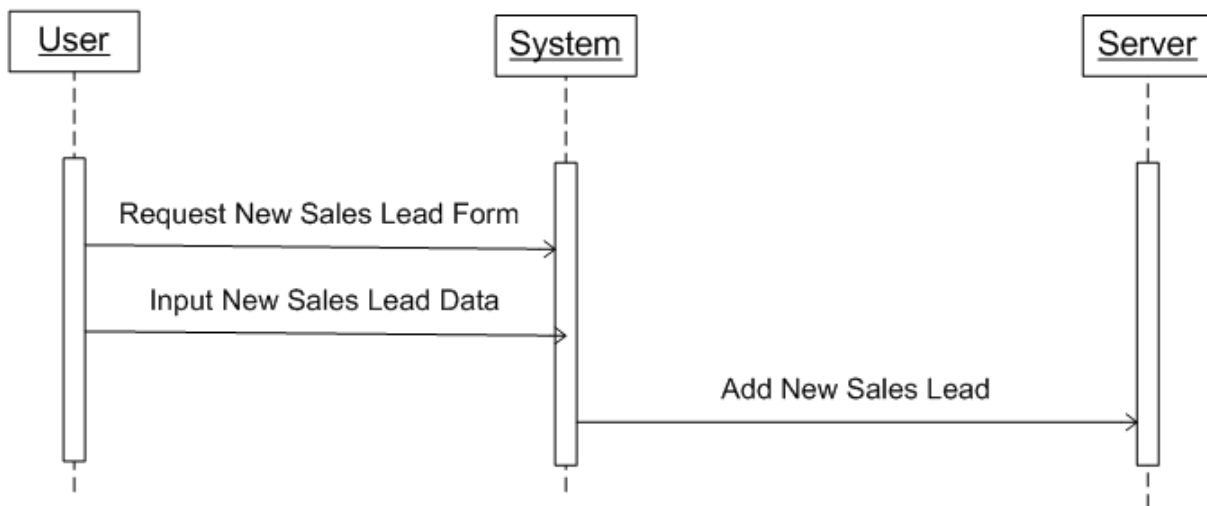
Add Customer



Edit Sales Lead

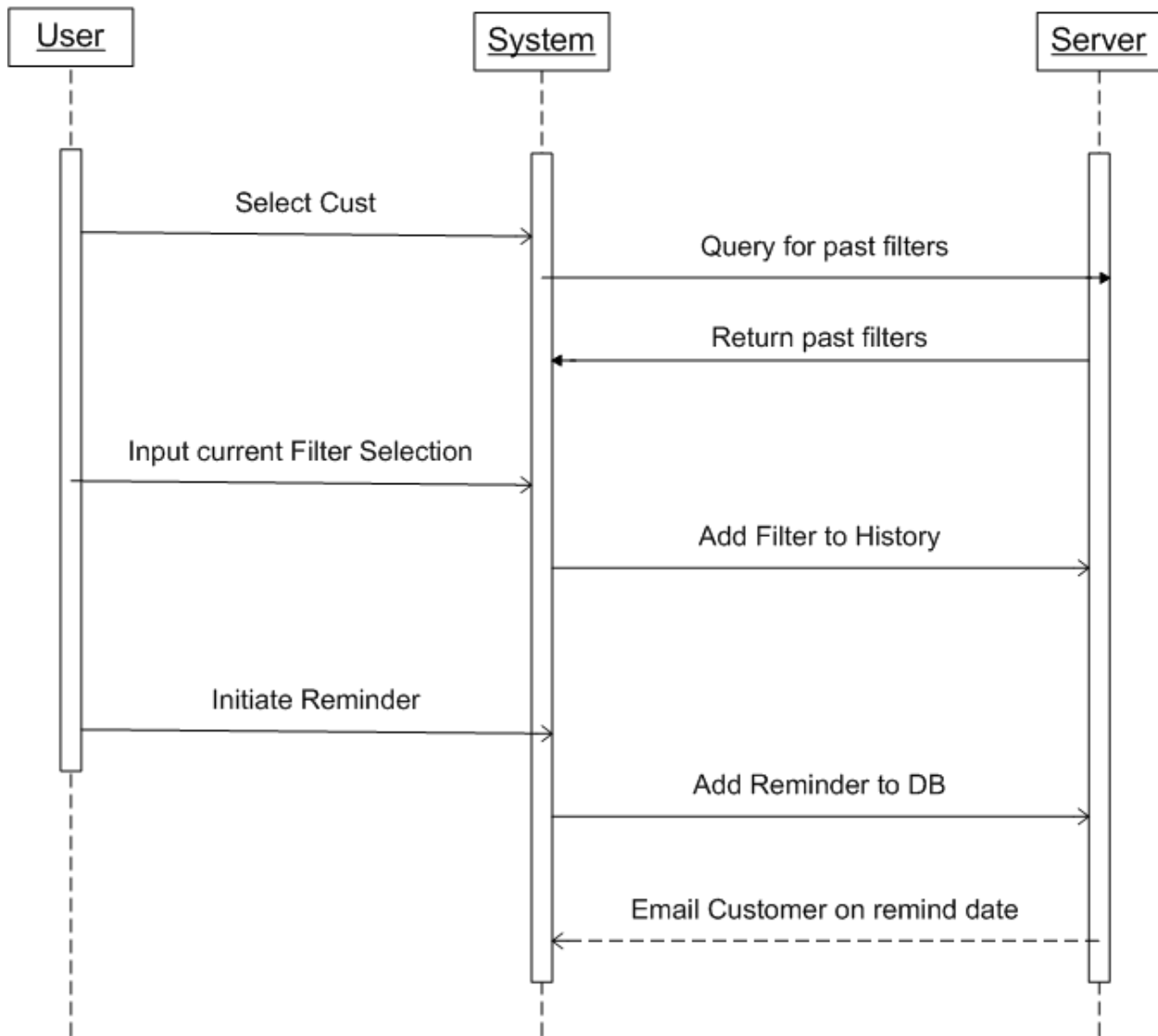


Add Sales Lead

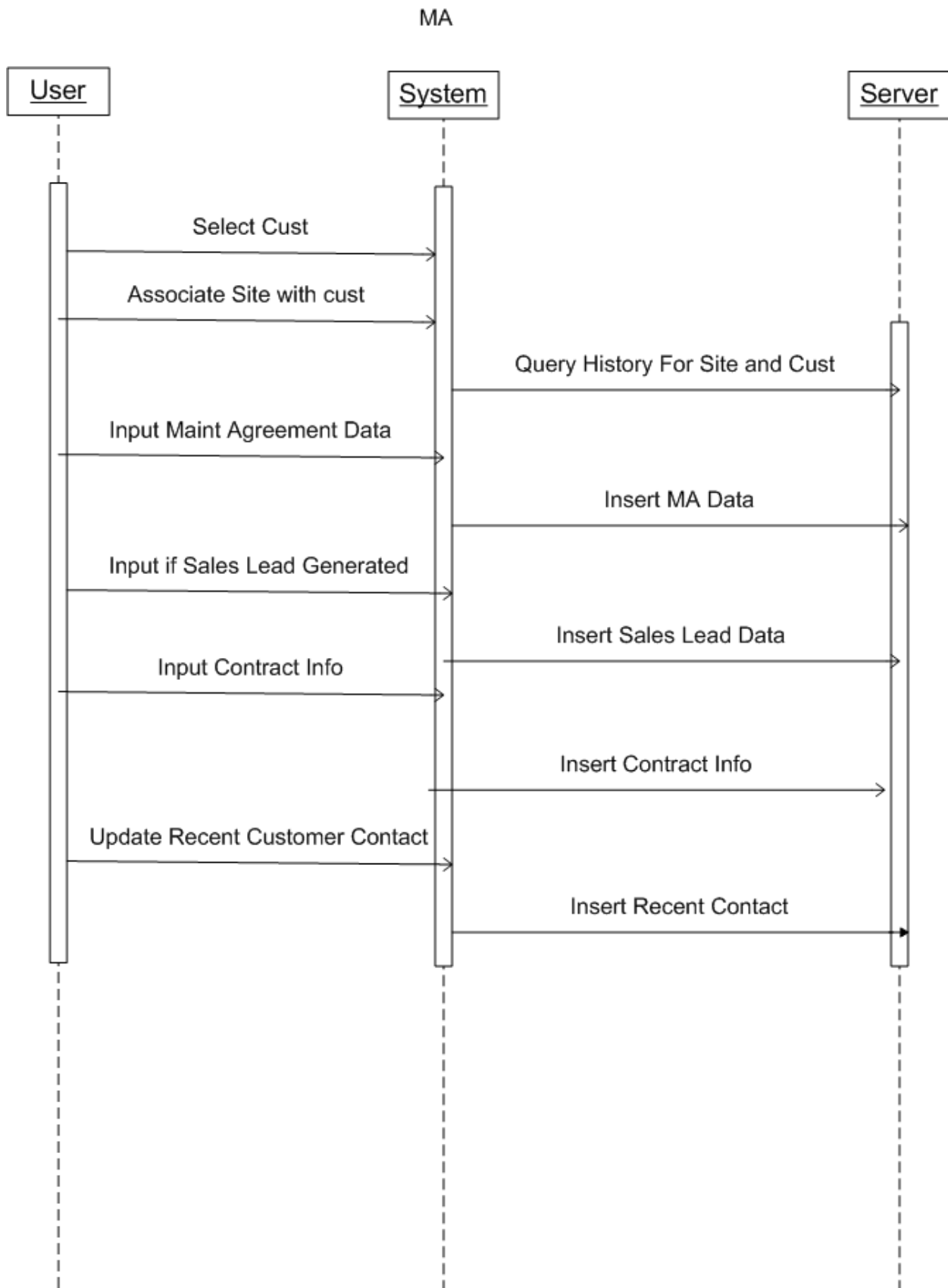


May 5, 2009

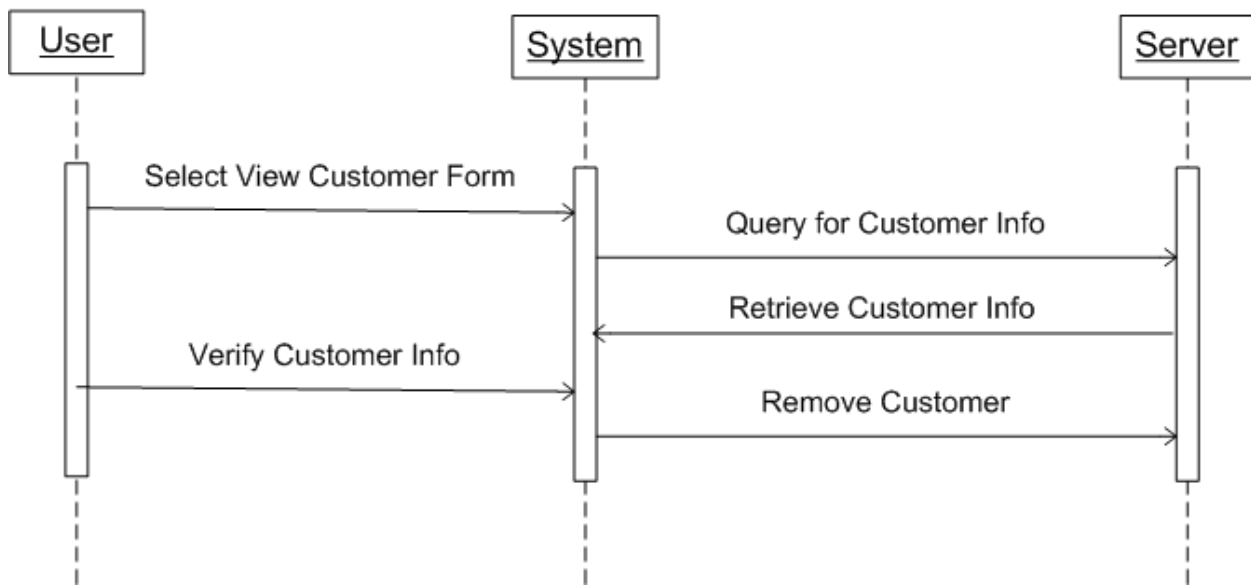
Filters



May 5, 2009

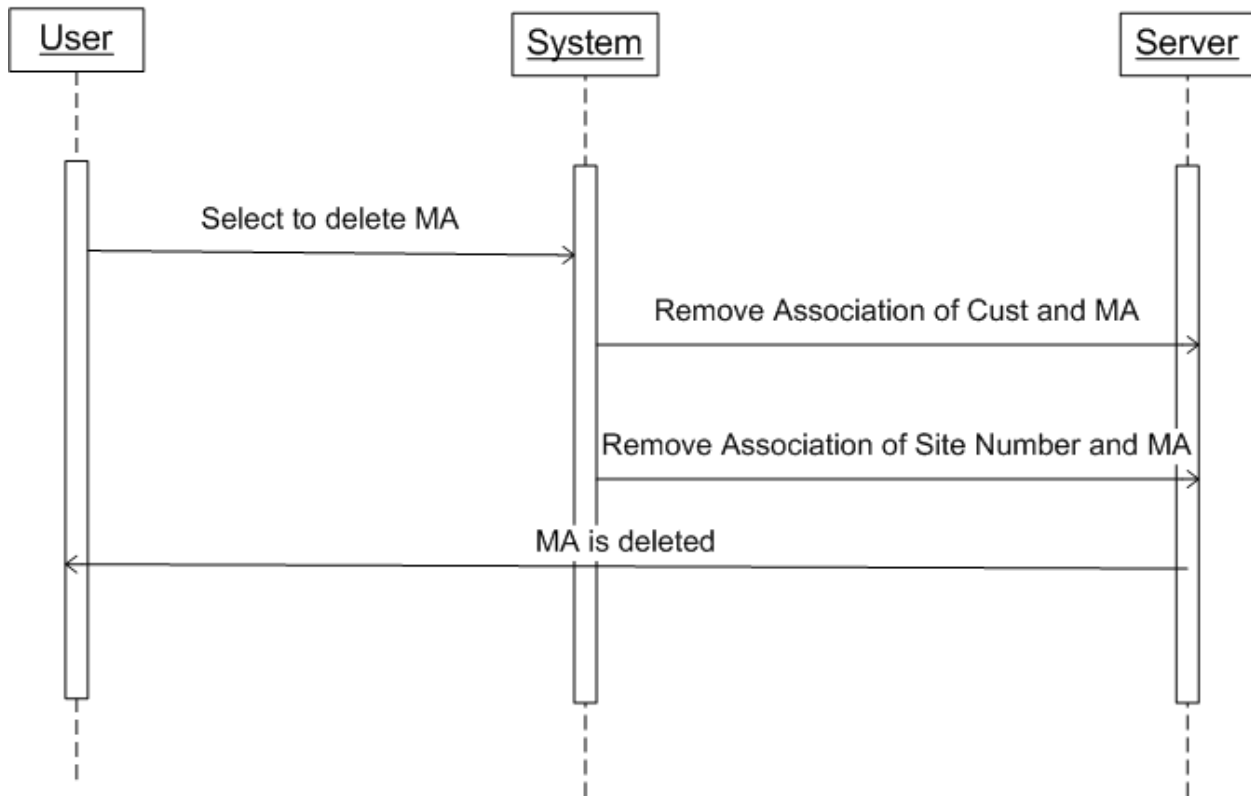


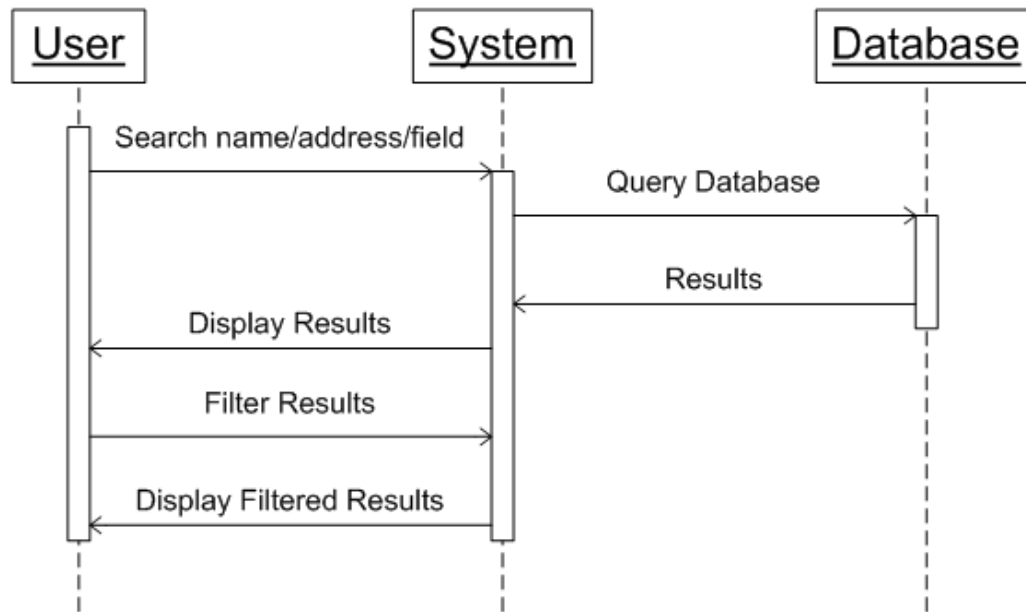
Remove Customer



May 5, 2009

Removing MA

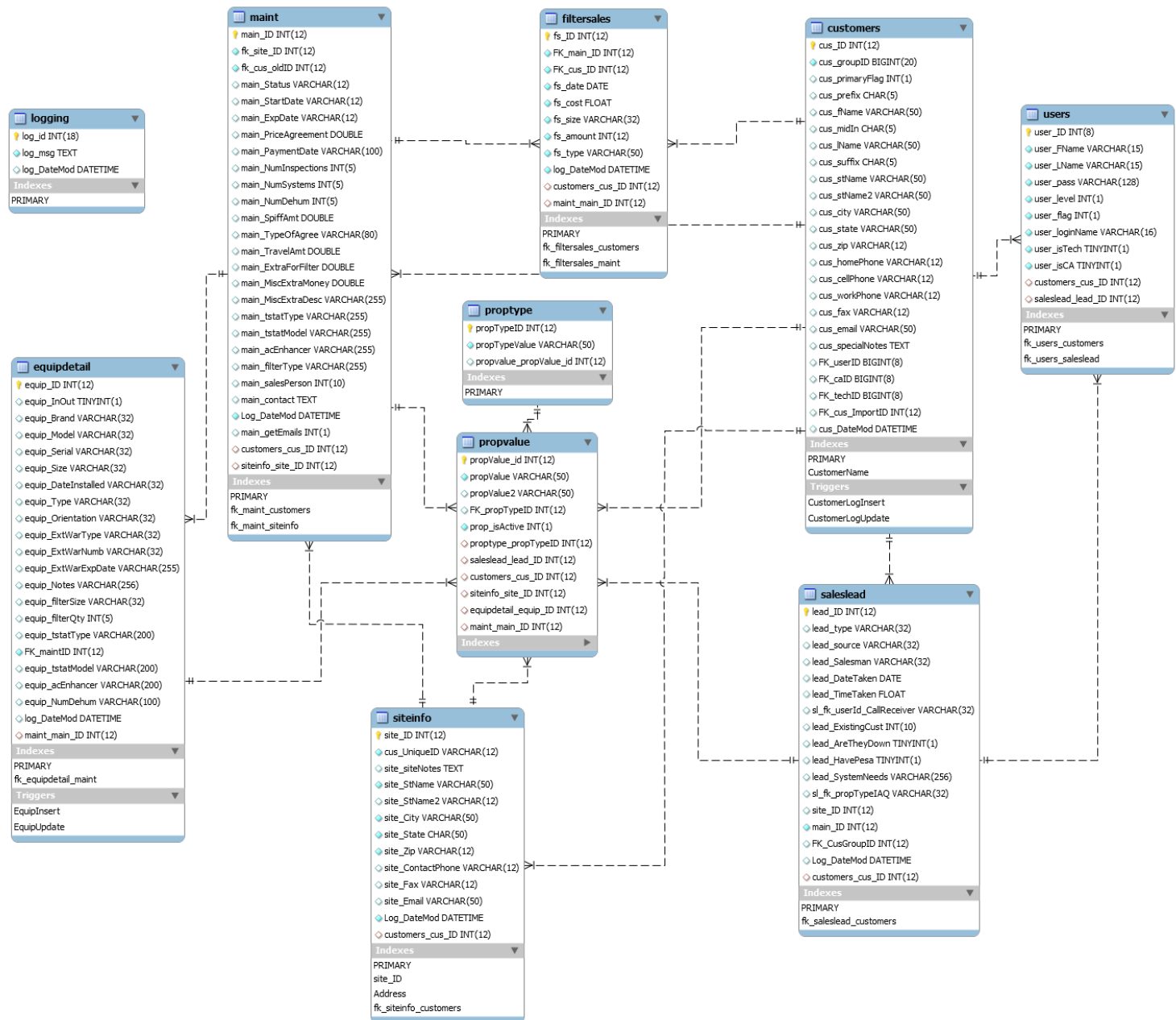




May 5, 2009

Database Diagram

Appendix E



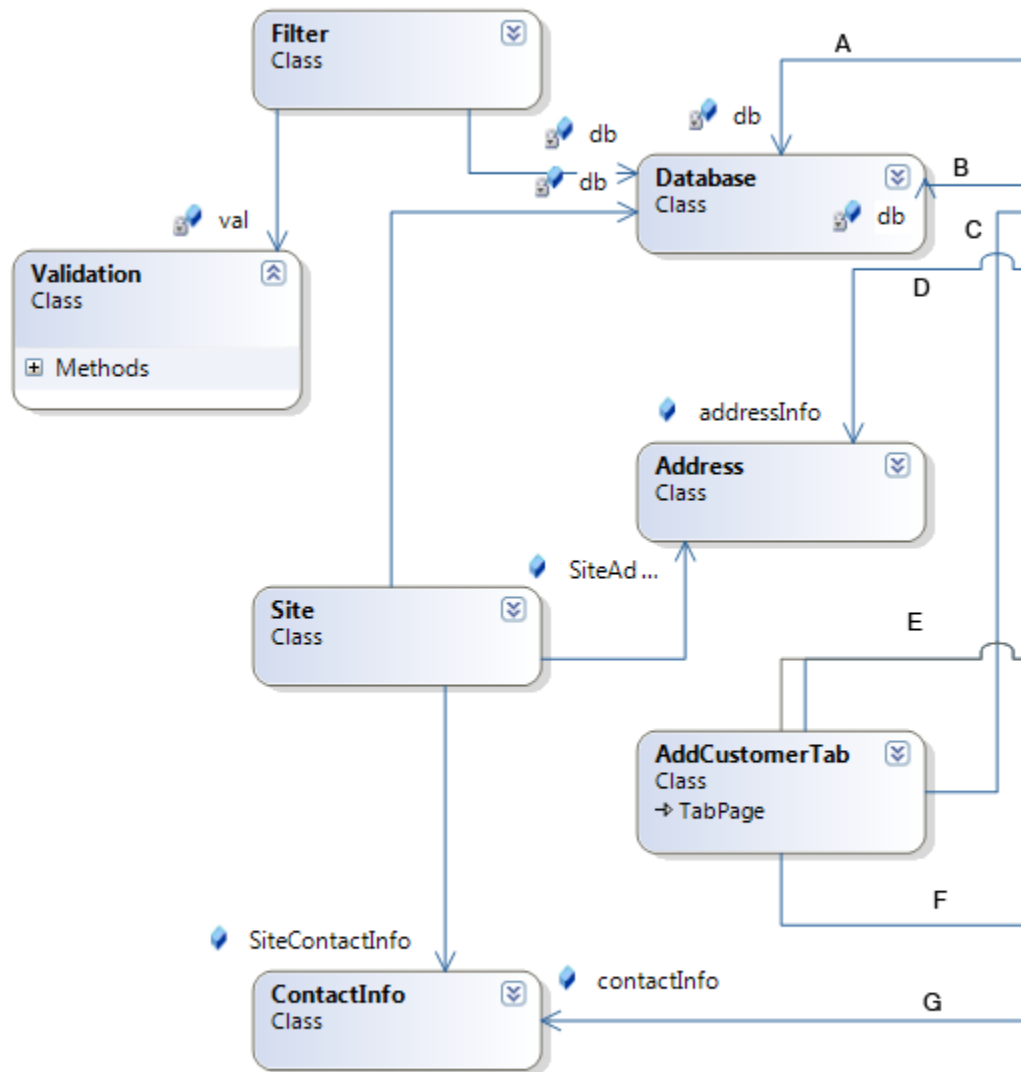
Class Diagrams

Appendix F

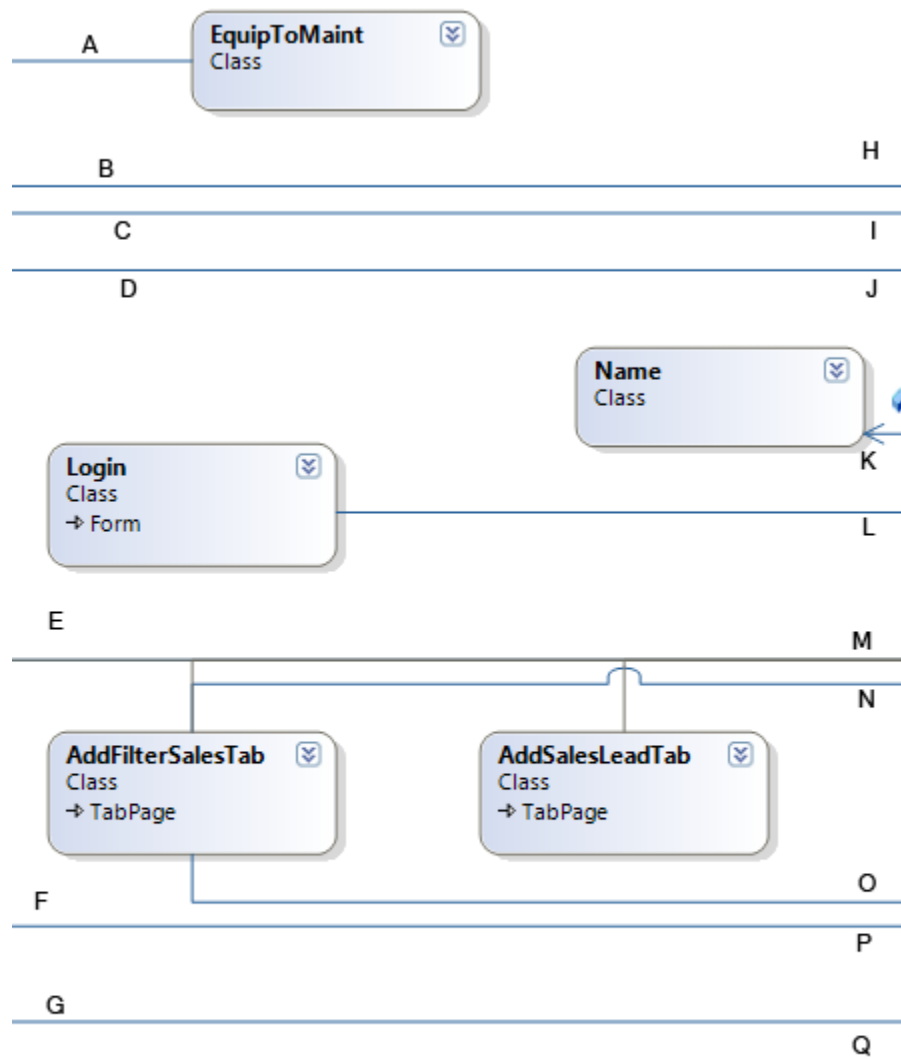
Please look at the file ClassRelations.png on the included CD to see the full class relationship image. Please note that the images below are not the actual class diagram, only the classes with their

May 5, 2009

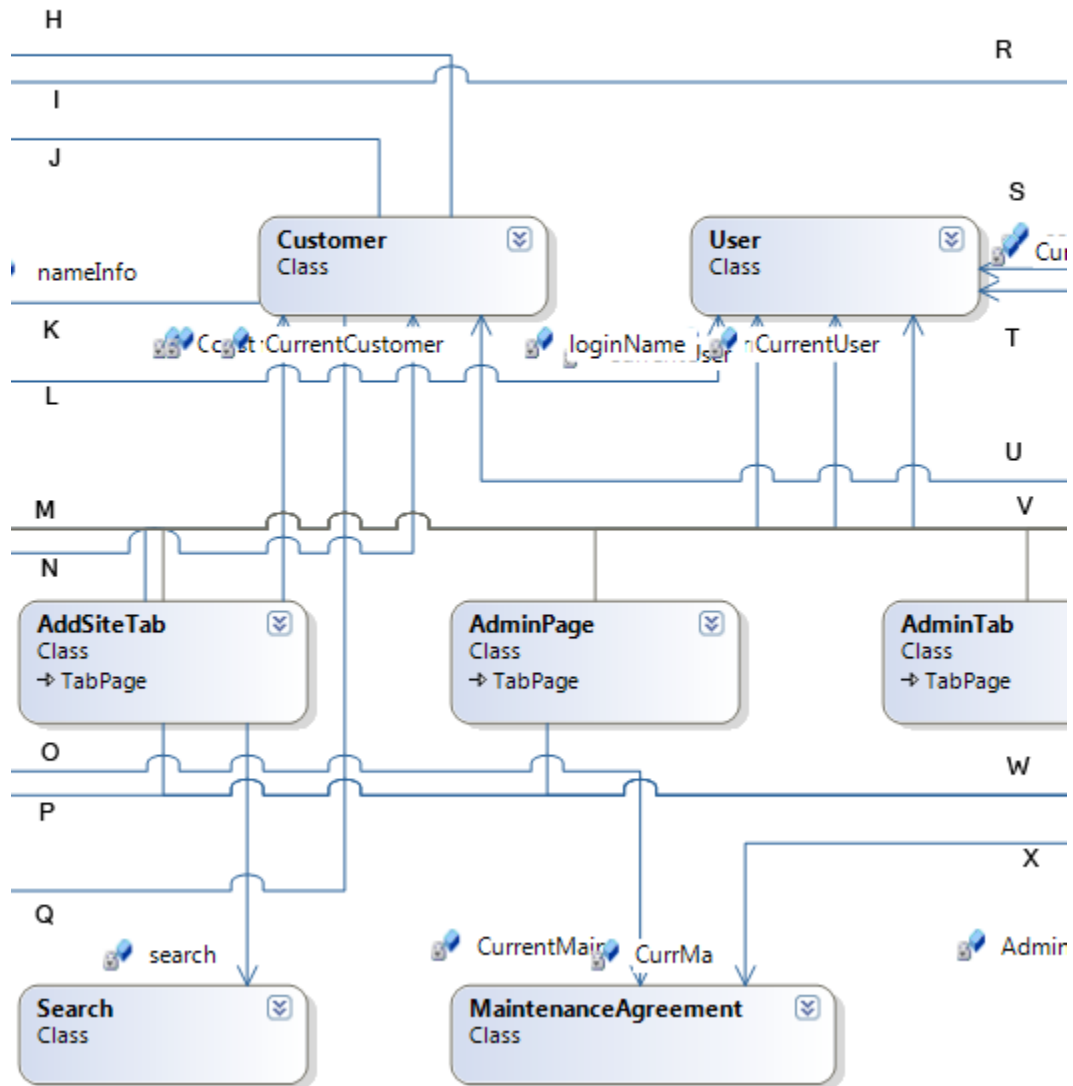
relations, in order to view the actual class diagram please open the file ClassDiagram.png located on the CD.



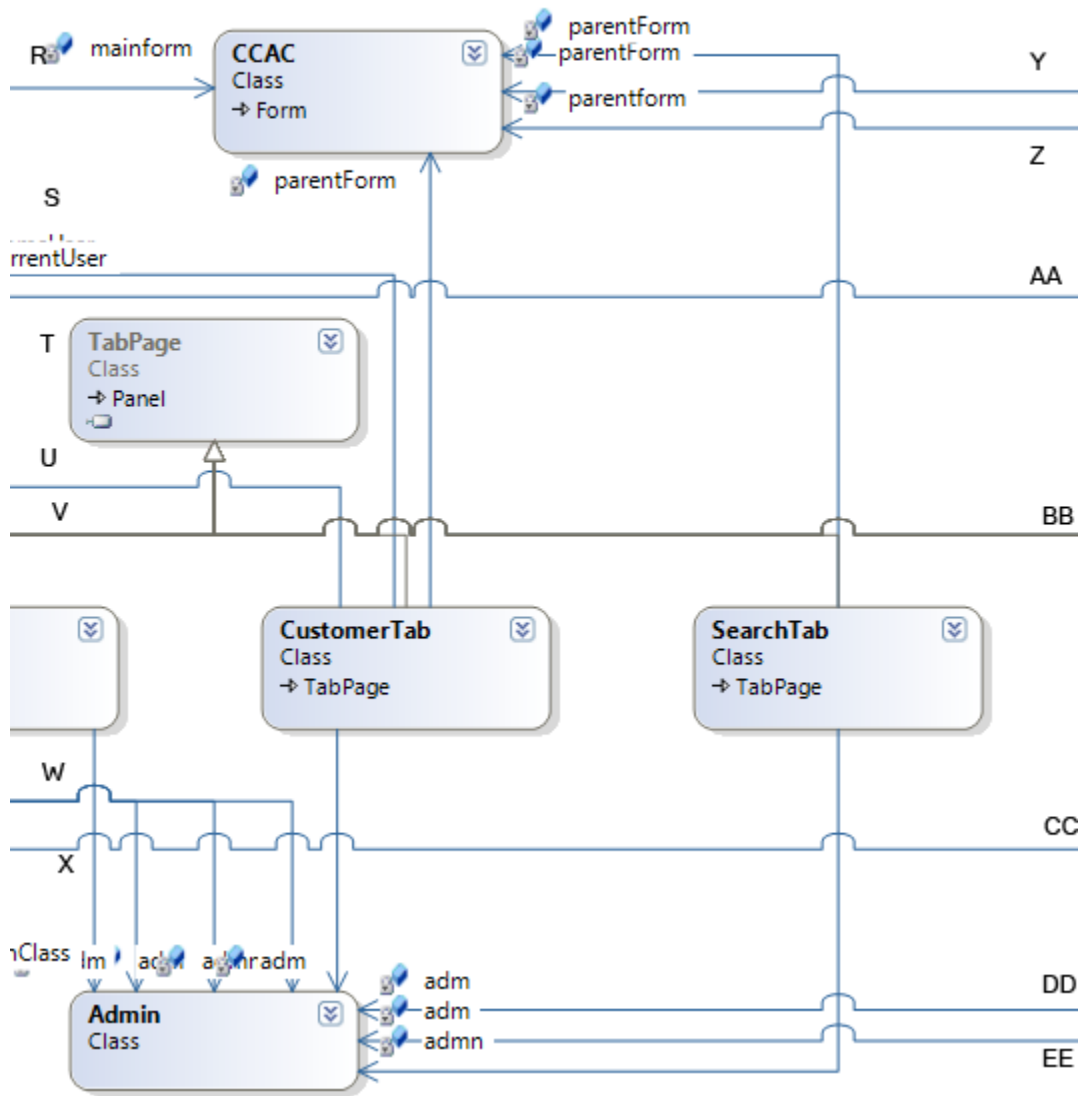
May 5, 2009



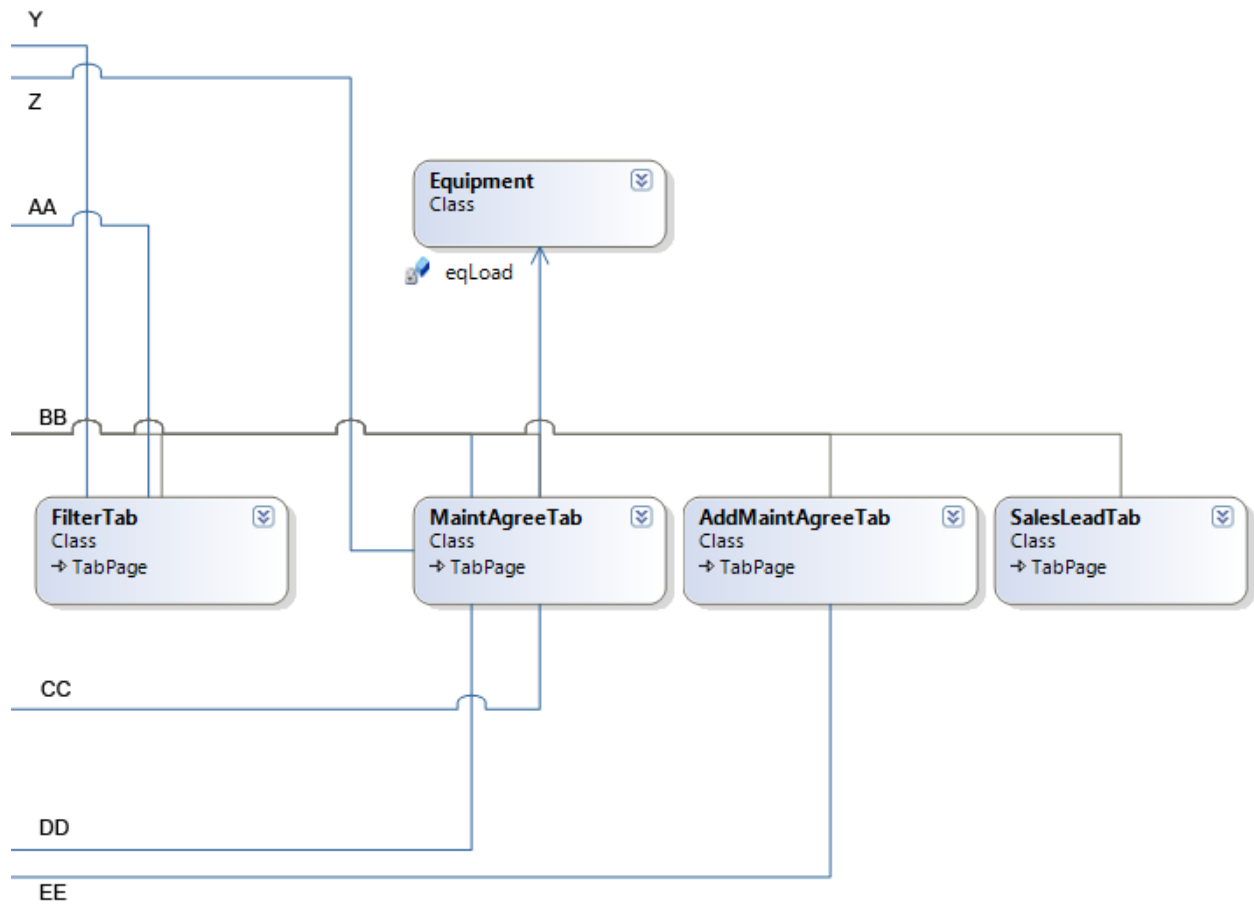
May 5, 2009



May 5, 2009



May 5, 2009



CCAC Capstone Project

May 5, 2009

